CIS3190 A4 Reflection

Conor Roberts #1056167

Compilation

Fortran

```
gfortran -Wall calce.f95 -o calce
```

Ada

```
gnatmake -Wall calce.adb
```

C

```
gcc -ansi -Wall -std=c99 -o calce calce.c
```

Python

```
python3 calce.py
```

Review

To begin this assignment, I worked to implement the algorithm in Python. I felt this was the best choice because I could leverage Python's high-level nature as well as my own personal familiarity with the language. This implementation proved very easy to accomplish. Once I had an idea of how the algorithm looked when written in modern code, I could better understand how to translate it into the lower-level languages. From there, I moved on to C, then to Ada, and finally to Fortran.

Translating the Python code into C was also a very easy task. These languages share a lot of similarities in their syntax. Where they differ heavily, however, is in variable declaration/allocation and file I/O. Due to my familiarity with C, this challenge was easily overcome. Ada and Fortran proved to be equally challenging due to their lack of similarity with the aforementioned languages. Both Ada and Fortran are very particular about string declaration and allocation.

Benefits & Limitations

Python
Benefits
- Intuitive syntax
- Plenty of available documentation

Limitations
- Lesser performance when compared to the other languages used for this assignment
- No strict typing

C
Benefits
- Better performance than Python
- More "modern" syntax when compared to Fortran and Ada
- Plenty of available documentation
- Strict typing

Limitations
- Errors at runtime can be non-descript

Ada
Benefits
- Unbounded strings
- List declaration from range
- Inline list initialization

Limitations
- Severe lack of documentation
- Unintuitive import syntax

Fortran
Benefits
- None

Limitations
- Everything