Software and Legacy Systems

A4: Comparing Languages
Ackermann's Function
Dominic Lee - 0823091

Task

To create a program that calculates the solution to the Ackermann's function given two integers. A non-recursive solution was built in python and recursive solutions built in C, FORTRAN 95 and Ada.

Overview

Having gained experienced in all of these languages, this program was successfully built for each one. The logic I initially used for the first language was essentially the same for each. All these languages supported sufficient structures to perform the task at hand. I found python the easiest language to develop in, as it supported some of the built in objects such as 'stacks', which were used in my non-recursive algorithms. Python also took the least lines code to perform the task.

C was the second easiest for me, as I have the most experience in this language. I also found FORTRAN 95 easy to develop in. Though Ada was my least favorite of the four, it was still fairly easy to develop in, as it had all the necessary structures and the same logic could be implemented.

The recursive functions all had faster process times than the non-recursive ones. FORTRAN 95 was observed to be the most efficient in its computation, as it had the fastest recursive and non-recursive process times.

Languages

C

I developed the first program in C, as I was most familiar with this one.
C allows for recursive and non-recursive functions, so this could sufficiently perform the task. I used an array-based stack, which was implemented in the non-recursive function. Relative to the other functions, C had pretty fast process times.

Python

Python, being a scripting language, made it the easiest to develop in for this particular task. Certain data type declarations were not necessary, as opposed to other languages. Python supports object oriented programming, so 'stacks' and stack operations were already built in. As I only needed to implement the non-recursive function in python, the built in features made it fairly easy and efficient. In terms of computational efficiency, python was the worst, as it had the slowest non-recursive computation times.

FORTRAN 95

FORTRAN 95 is known to be very efficient in making scientific and mathematical computations. This was definitely the case, as the process times for both recursive and non-recursive functions were the fastest of the four languages. Implementing the logic for the recursive and non-recursive functions were fairly straight forward, as it did not deviate from the logic I used for previous programs.

Ada

Ada supported both recursive and non-recursive functions. Ada also supported the use of packages, which contained stack objects and operations as well. Though my least favorite, it was still fairly easy to develop in. I didn’t really find any benefits to using this language, however. Process times were a bit slower than that of observed in C.

Conclusion

All languages were fairly sufficient in performing the task. In terms of ease of development, python or C would be my top choice. If a larger number of computations would have to be made in the future, where process time is critical, however, Fortran would probably be best choice.