Braelyn Rotman
ID: 1006740

Assignment 4 Question 3 – Russian Peasant Multiplication

Reflection Report

Below are tables for each of the multiplication functions to compare the execution times of each language. The execution of each function was timed rather than the entire program for the most accuracy.

Recursive Function Execution Times (in milliseconds)

	Fortran	Ada	C
6 x 7	0.0070005330	0.004	0.001
37 x 86	0.00600004569	0.004	0.001
93 x 734	0.00500003807	0.005	0.001
426 x 843	0.00400003046	0.003	0.001
639 x 3047	0.0140001066	0.004	0.001
5039 x 2930	0.00600004569	0.004	0.002
3974 x 18434	0.00600004569	0.004	0.001
34048 x 47395	0.00600004569	0.004	0.001
43085 x 285748	N/A	0.005	N/A

Non-Recursive Function Execution Times (in milliseconds)

	Fortran	Ada	C
6 x 7	0.00300025567	0.000	0.000
37 x 86	0.00300002284	0.001	0.000
93 x 734	0.00300002284	0.001	0.001
426 x 843	0.00299979001	0.001	0.001
639 x 3047	0.00300025567	0.001	0.000
5039 x 2930	0.00300002284	0.001	0.001
3974 x 18434	0.00300002284	0.001	0.001
34048 x 47395	0.00299979001	0.001	0.000
43085 x 285748	N/A	0.001	N/A

Braelyn Rotman
ID: 1006740

A Note on Efficiency

Fortran was clearly the slowest language for both functions and C appears to be the fastest language. However, I noticed when testing my program in C the execution time would come out as 0ms many times which makes me wonder how accurate the timer is. Unlike Fortran, the Ada and C timers did not give as accurate data making it difficult to compare the languages, specifically in the non-recursive function results. Ada and C had the same results for all my tests. It is possible that Ada was more efficient than C however due to the instantaneous execution the test results show C has the more efficient language.

Multiplying Large Numbers

The difference in execution time for multiplying small numbers and large numbers was insignificant throughout all 3 languages. Therefore, in terms of efficiency no language was significantly faster at handling large numbers. However, looking at the test results above, Ada was able to handle larger products than C and Fortran. The larger numbers caused overflow and showed a negative product in the output.

Implementation Experience

The algorithm was simple enough that I didn’t have a hard time implementing it in any of the languages. Since all 3 languages allowed the use of recursion, loops, and case statements I only had to change the syntax between them. Honestly, the most difficult part of this assignment was formatting the output to be something readable and implementing the timer functions. Despite being the language I am most comfortable with, C took the longest time. I was unable to find a way to time the functions with more accuracy than 0.001 milliseconds. I’m still uncertain if this was due to the timer lacking accuracy or the functions were simply running instantly. Ada was similar with the non-recursive function which made it difficult to compare the two. I considered timing the entire program for a comparison however since I take input from the keyboard the results would be solely dependant on how fast the user types and would be unreliable for estimating efficiency. The execution time would have also been affected by the different timers and the calculations needed to output the millisecond execution.