Course Project : Magic Square Parser

Aryaman Khandelwal

CS 2001: Programming Language Principles and Design

Professor Kanchi Gopinath

May 25th , 2023

In the POPL project, I used Python, Julia, and Swift as the three languages in which I coded the Magic Square matrix problem.

The reason for choosing these three languages only, from the list of all the languages that were given to us, is because of the following reasons:

- Python: I have been coding in Python for the last three semesters. This has made me fluent and well-versed in the nitty-gritty of the language, and I had confidence that writing code for the problem in this language would be relatively easy compared to the other languages.
- Julia: The syntax of the language is very similar to Python. Moreover, many concepts and programming paradigms are similar between Python and Julia, which made me pick Julia as my second language.
- Swift: I chose Swift because I have a MacOS and wanted to experience the language. Moreover, it is used for developing mobile applications, which is my area of interest.

I started the project by picking Python as my first language. It took me a lot of time to complete the problem in Python, as I was decoding the logic and coding simultaneously. Since it was the first time I was attempting the problem, the majority of my time went into logic construction, operations individual functions will perform, and considering all the border cases.

One aspect of Python that posed a challenge was the regex function. I found it difficult to understand the regex function, and getting the desired output took me a long time.

The flexibility of the types of data structures in Python, and the ease of conversion from one type to another, eased the implementation of my logic.

My experience with Julia was good. My fluency in Python helped me a lot as the syntax was similar. I would face general hiccups due to different syntax, but I solved these by referring to the official documentation. However, the one aspect of Julia which troubled me a lot was that indexing starts from 1 and not zero. Due to this, I faced the error of index out of bounds several times. Consequently, I had to alter some of the functions' logic and pay extra attention to the code to avoid the index out-of-bound error.

The following error I faced in Julia was of the isalpha() function, which checks whether it is an alphabet. This function was not being recognized. However, VSC suggested it. On many websites, they were using this function without facing any difficulty. Therefore, I had to use an alternate way: isascii () and isletter().

Coding in Swift took me the most amount of time. The primary reason was the significant difference in syntax between Swift and Python, Julia. The next and primary reason was because of its type safety and labelling of types. While type safety helps in type checking, it posed as a big challenge to me. I often faced a type error, like [String] cannot be converted to String.

To ensure strict type checking, Swift implements labelling and declaring of variables. It was absent in both Python and Julia and was a massive pain for me as I ran into many such errors.

Initially, I was unable to understand the source of the error. But, after searching on the net, I got an idea.

In all the three languages, the logic is same. The logic of all the functions is same across all the three languages, the difference lies in their syntax and implementation. In each language, I have made around 15 functions. Each function performs an atomic operation, that is used in the process of checking whether a matrix is a magic square matrix or not. Comments have been given in each program, to understand the objective of each function.

To sum up, I had a great learning experience in all three languages. I understood different concepts of programming languages, like dynamic typing, strict typing, and type safety. Finding out different ways of handling errors was also a critical insight from the project. Thinking about all the corner cases and devising logic for each enhanced my thinking ability. The project also taught me how different languages are designed and implemented differently. Python gave me great comfort and flexibility, whereas Swift restricted me and focused more on safety.