

Lab 2

In this lab, we created a program to convert prefix to postfix notation. This was very similar to what we did in Lab 1, however we used recursion to solve the problem.

We recurse through the input expression. First, we check whether the given prefix location at index i is an operator or not. When an operator exists, we check that two operands exist or not. If so, we properly pop and add to an auxiliary variable that is then pushed into our stack. If not, we then set the `is_valid` to `False`. When we get valid operands, we push to our stack. If we find something incorrect, we raise an error / exception.

In this case we do not raise an exception as we want the program to continue with the entire input. We have several options on how to go about this but find returning a string as an error and popping the stack will allow us to iterate over the entire input file and return the results as a dictionary to the end-user.

We have also created a container in a Dockerfile to run the program. We expect that containers will be used via Kubernetes or another tool like airflow to run the program continuously.

Unit tests have been written in the test's directory and passed based on both stack and recursion.

Thanks
Jordan