# Approach

My approach for this project was first to run the provided skeleton code and observe the changes made in the code to allow for our language to be interpreted. I then took note of what was missing in this code versus what we did in our last project and then followed along with the provided approach document. It was important for this project to not just combine code from the last project with the provided code as many errors show at once and sifting through the errors one by one is not the most efficient way of attacking this project.

# Testing

There are many things to test with this project which includes:

* Testing of arithmetic operators
* Testing of logical operators
* Testing of relational operators
* Testing of parameters
* Testing of variables
* Testing of control statements and reductions
* Testing of literals

### Test 1

In this test I take in 4 parameters, have 3 variables, and use a case control statement to output the result calculated for each variable. This tests that parameters of all types are allowed, tests all arithmetic and relational operators, tests all literals, and tests a non-nested case control statement, that is controlled by the 4th parameter.

* 1. – arithmetic output:

Text

Description automatically generated

* 1. – real output:

Text

Description automatically generated

* 1. - Boolean output:

Text

Description automatically generated

### Test 2

This test tests that the program interprets correctly with no optional variables or parameters.

2.1 – 1st and 2nd if statement are true:

Text

Description automatically generated

2.2 – 1st if true, second if not true