

## Question 2

The task for question two was to implement other functions commonly found on a scientific calculator and to write a summary of the functions added. For this question I went with the most common functions I used on my scientific calculator when taking calculus: Sine, Cosine, Tangent, and Square Root.

All of these functions were implemented the exact same way since the format looked like the following: **Func Exp** however they have different precedence in the order of operations.

For the trigonometric functions I modified both the Bison specification file and the Flex definition file. Adding to the Flex definition file was the following three lines:

```
"sin" return SIN;  
"cos" return COS;  
"tan" return TAN;
```

This is basically the token definitions that we end up using in the Bison specification file. The modifications made to the spec file were as follows. Tokens were added in after the **POWER** token because the order of operations require trigonometric functions to be evaluated before exponents. Then we add that SIN, COS, and TAN are all left associative by adding the following line:

```
%left SIN COS TAN
```

Finally we need to add the rules of the expression. These are all added after the **POWER** expression as well to keep precedence. Since SIN, COS, and TAN each only take one argument we need to modify how the expression looks from **Exp Func Exp** to just **Func Exp** as the following lines demonstrate:

```
SIN Expression { $$=sin($2); }  
COS Expression { $$=cos($2); }  
TAN Expression { $$=tan($2); }
```

Then to implement the Square Root function was identical in all ways to the trigonometric functions because it also takes only a single argument as well as it is left associative. The only difference between the two is the precedence in order of operations. I was placed before **POWER** because it can be evaluated at the same time, just before, or just after. Another way to express  $\text{sqrt}(4)$  is  $4^{\frac{1}{2}}$  which is just an exponent.

Now that I understand how to add new functions I could easily add an array of additional functions but I think four proves my understanding good enough for now.

## REFERENCES

<http://www-h.eng.cam.ac.uk/help/tpl/languages/flexbison/>