For Lab 3 I decided to work with Lex and Yacc to create a programming language to solve the sample grade calculation problem which was.

*Example of a use case of creating a calculator to calculate your final grade.*

*In this design, entry of category type of activity (L for lab, Q for quiz, R for reflection) with the numbers afterwards the list of numbers to sum for that category. T to retrieve the total of presented categories.*

When I started this project in Lab 2 I at first had some difficulty programming with Lex and Yacc due to the Linux terminal work flow. While it is still something I still need to learn, it did become a bit easier for me with time.

I now have a working program. I am however not proud of everything that the program does however. This is because I had to remove some of the features, I originally was aiming to have in order to get the program to work better.

One example of this was allowing the program to read in grades with multiple digits.

Ideally, I wanted the language to understand

Q 12 9 9 9 9

Meant 12 followed by four 9’s. This is not how it was interpreted though.

It instead interpreted it as

1

2

9

9

99

In order to fix this problem, I reduced the features so that it became

Letter single-digit-number single-digit-number single-digit-number single-digit-number single-digit-number

Or

Q 1 1 1 1 1

While a working solution this limits the scope of the program to grades that have a max score of 9 which will work for some cases but not others.

Honestly, I can explain the solution to the problem quite easily especially with Regular Expressions but it was interesting to learn that even though the concept seems simple and foolproof the actual execution could still have problems.

Another issue with this is if I accidentally enter 6 numbers like

Q 1 2 3 4 5 5

It uses the first 5 numbers in the grade calculation.

Here I would ideally want it to reject the entire input.

A similar case happens when you do

T 5

T means you want to get a total.

Here 5 does nothing but typing this will still provide you with the total.

Again, I would prefer it to reject this input as it is not T alone.

While it is fine for it to do this, I would prefer any deviance format to be a red flag.

* Which utility did you select?

Lex and Yacc

* Why did you choose that utility? What were some useful resources to learn how to use that program?

They were the recommended utilities. Online resources and YouTube videos were use useful.

Also, the notes from Professor Weiss were very useful in figuring out how to construct both files.

* Was there any difficulty or ease with using the terminal environment?

As mentioned before I am a novice when it comes to terminals but it did get easier with time.

* What kind of language are you designing?

A language to calculate grades and store them as well as a total.

I found this exercise with Lex and Yacc to be very interesting. While At their face value I think the Lex and Yacc values come across as hard to read or intimidating when you take a deeper look it does become simpler. To be making a program with Lex and Yacc is like building a real language or a secret code. What you are describing is the different pieces and how they relate together to form actions.

Also, despite the troubles I thought this was a very fruitful lab. I may even consider working more on this language as a side project so I can come to understand more about programming languages and compilers and interpreters.