For this Lab I attempted 2 things.

The first was programs using the Mips programming language and the MARS IDE. The reason I decided to do this was because I am not completely used to using Linux and Command Line so I wanted something that could take advantage of some of what I already know. In my Computer Architecture class last year, we were introduced to some of MIPS and used the MARS IDE. So, by having some familiarity I felt that it would be a good backup to have since I thought I would probably struggle with Yacc and Lex.

The first program I made was a program that asks a user to enter a number and calculates the sum of all numbers from 1 to that number. I did it by using loops. Thinking back on it using the formula (1/2)\*((N+1) \* N) would have probably been better. Also building off of this program I made one that calculates the factorial of a number entered by the user. This was very easy to create as I just had to slightly modify the previous code. I thought this would be a good warm up since I have not worked with the language or IDE for some time. By working with it again I remembered that Assembly language can be a bit tricky to work with since what is just 1 line in a high-level language can take multiple lines. Also, since most of the keywords are very short and sometimes acronyms, I had to use a cheat sheet to look up the functions I needed.

After I brushed up a bit, I decided to make another program. This was a program that counts the number of 0’s in the binary representation of a word. This was based on an assignment I remember doing the previous year. One problem I faced with this part was figuring out how to actually go into the data to check the bits. I found this a bit difficult as I do not think the description of the function of AND and SLL (shift logical left) were that clear. I did eventually figure it out though.

* Which utility did you select?

Mips and the MARS IDE

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

I chose it because I have a little familiarity with it. To me what was most useful were my old notes from CISC 3310 Computer Architecture.

The second thing that I attempted to do was create a programming language to solve the example problem of grade calculation. The reason I decided to do this was because I appreciated the direction the example provided. Without it I would have no clue on where to start. The first thing I did was to try and set up my own short language using an online tutorial. After having no luck, I decided to clone Professor Chuang’s example repository directly onto the school Linux server and follow her instructions on how to get it up and running. This worked perfectly. So, I then decided that I would edit and build on that to make it a little easier on myself.

One problem that I had with trying to make my own language is that there seemed to be so much symbols and syntax that I had to use that I was unfamiliar with it. While it might be difficult though this is something that I want to learn. I might even try to create a more expansive programming language than the one for the simple problem in my own time because it seems like a really great opportunity. One problem I see in doing so is that it seems like a lot of work but many worthwhile things are. Also, since it seemed like a lot of work to get it to handle the problem, I came up with an idea that would be a cheap method to solve it but would probably be easy. My idea was that I can reduce the input of

R 5 4 4 5 5

And so on to a regex expression. This meant there might be a way for me to be able to solve the problem using only a lex file and the default yacc file from the music example.

The regex expression I came up with was

([Qq](,[0-5]){5})

This specifies to cath the sequence of

Q,3,3,3,3,3

Or Q or q followed by 5 commas and 5 numbers.

The reason I had to change this was because I was having trouble getting the regex expression to recognize spaces.

Using a simple print I could show that

Q,2,2,2,2,2 would cause Quiz to be printed

But

Q 2 2 2 2 2

Would cause an error. I am not exactly sure why that happened but I will try to fix it since I consider changing the delimiter to a comma to be cheating a little bit. Another problem I had was with working with the numbers so I could sum them up and store them to a variable. I do believe it is possible but I will have to read more on how to use Lex and Yacc. Also, it might be the case that doing it my way even though it sounds easier is more difficult that creating the language properly. That is why I also want to try solving it the proper way.

* 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 recommneded untilities. Online resources and youtube videos were use useful.

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

I did have a bit of a difficult time. I chalk it up to me not knowing much about how to use command line. For me I like the ease of coding in an IDE but learning command line is definitely a skill I need to master. Another thing I should work on more is how to use GitHub. This is because for the most part I only push the final product to GitHub. This is not good as I have actually had times where I have working code and then change it thinking I optimized it and then the code breaks and sometimes I do not remember what the code was. Version control could definitely help me with this. My method of getting around this is by copying the working code into another file but having this stuff on GitHub would be much better. One problem I face in doing this though is that I do not find GitHub particularly user friendly especially to people who do not know that they are doing. But I guess that is just another thing that I need to learn.

* What kind of language are you designing?

I am designing a language to calculate grades. I am trying to design it in 2 ways. The first is the proper way. The second is to simplify the solution by using regex.

One thing I can say though is that by working with MIPS, Lex and Yacc I definitely have more appreciation for the people who make IDEs and languages. It is a tough job and makes our lives much easier. Also, while I do find the idea of creating my own language easier, I do feel like I am more comfortable with the higher-level languages and that it might be better if I stick to them. Also, I will say the info on programming languages, how they function and how they make them will probably be the most interesting thing I have learned this semester and I probably will look into it more.