**WORK LOG**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Language | Hours Spent | Tasks/ Accomplishments / Issues |
| 2-Feb | ForTran | 2 | Intro to ForTran. Followed a bunch of tutorials like Hello World. Surprisingly not too difficult looking or confusing. I already have an idea of how I will handle ForTran. |
| 3-Feb | ForTran | 3 | Began to do the program. I think I got the computation down but there is an issue with these characters not being able to print. The debug shows that input is encrypted correctly when under breakpoint. |
| 10-Feb | ForTran | 2 | Spent more time trying to figure out this output issue. Nothing is working. The input is right, the encryption AND decryption are right, but its crashing on output. |
| 15-Feb | ForTran | 2.5 | Spent even more time trying to figure out issue, may move onto another language or put it on hold for a bit. Physics takes up 12 hours of work a week anyway. I tried to change character lengths with trim. Not working still. Driving me up a wall |
| 22-Feb | ForTran | 4 | Figured out the issue with printing out strings after encryption/decryption in Fortran. You have to initialize string characters as well, just like other languages. Who would've guessed? Also began to try to organize program. |
| 10-Mar | Pascall | 3 | Started and~ finished Pascall. I was shocked at how simple it writes, but even more so how almost exactly the same method as Fortran is used. However, I ran into an error when trying to figure out lower case letters. Instead of killing myself over it (1 hour at least), I just made everything uppercase. Very successful. |
| 16-Mar | Scala | 3 | Started Scala. Few intro tutorials, and I kind of realized its similar to Java. But very weird and annoying to compile/run in the command prompt. I tried to program a bit for the Cipher but I feel like there may be complications compared to ForTran and Pascall. Unlike Java, it does not have the easy methods to just manipulate the strings or in Fortran with the modulo(iachar()) |
| 1-Apil | ML | 1 | Started ML instead of finishing Scala. I'm taking ML for Language Study and found that it isn't too hard. Its figuring out how to match the cases that took the large part of the hour I spent on it. Still not done |
| 5-April | ML & Scala | 2.5 | Finished BOTH ML and Scala. After tinkering around, I got ML to work after figuring out List.tabulate. In Haskell, you can do lists [0..5]. ML doesn’t have this in that form, but rather a built in function. Annoying, but I did it. Scala I figured out the mod.size for the modular function so it let me do the map function which checks the cases for the string. Also annoying! BUT ITS FINALLY DONE! |
| 6-April | Erlang | 1.5 | Really dislike this language. Its like pulling teeth. Everything so far was pretty decent and uniform when it comes to the encrypt function. Erlang, however, is not and I have no idea how to tackle this hurdle. The issue comes from using the convert function that the encrypt function calls after using a map. |
| 7-April | Erlang | 5 | After completing Erlang, I’ve learned something. Never go near it again. I ended up having to use band – a bitwise operation X band Y bitwise and. We had used a similar bitwise operator in Assembly, and I just hate this kind of thing. I can never wrap my head around the bitwise operations. Once I had that, it was a matter of trial and error until the math came together. |
| 13-April | Lisp | 3 | I actually had a good time with Lisp to my surprise. I completed it in one sitting. I read a tutorial website on the basics and realized there is nothing to crazy about it. I used a very similar map implementation as ML in order to get the convert function right. Once I got that to work, it was just a matter of figuring out the solve function. Which after first trying to do a funky map to do it, I realized lisp has a built in loop! I would’ve had this done in 2 hours flat if I knew that earlier. |
| 20-April | COBOL | 5 | I thought I would’ve been able to handle COBOL. I heard it was a big to-do, but I thought it was going to be a push over in the end. I was horribly wrong and naïve. This language is like Assembly on crack. It is obnoxious, long, and throws the most useless errors I’ve ever seen. I couldn’t even get anything done. My program doesn’t have a single line of valid code that actually does anything and I’m already 5 hours in. My roommate was actually getting scared at how mad and loud I was about it. It probably was augmented by the terrible Physics class I’m taking, but that’s another story. |
| 21-April | COBOL | 2 | Getting down to the wire here and panicking fast. I know I still have another week left, but I don’t know when I can actually got this done. I’ve asked other people what they have done and it seems even the people who have it done don’t quite understand what the bloody hell they did. Some of their solutions are actually manual implementations that just check against EVERY possible cipher from x to y and just switch characters as such. There MUST be another practical implementation to this. I’ve gotten about as far as input now and still getting obnoxious errors that don’t help. Periods are the worst. |
| 22-April | COBOL | 3.5 | More rage and anger. Admittedly at this point I’ve been asking for help and getting Travis to help just to get me off the ground. But I really dislike his implementation, its very manual. Not the most inefficient implementation I’ve seen, but its just weird. I do like how COBOL has this “INSPECT” function that magically sorts through a string and changes it. Very easy to convert from lower case to capital letters. |
| 23-April | COBOL | 3 | I THINK I have encrypt down. I finally figured out how to do it, and using an online compiler it looks like it works. However, how to get decrypt to work is another issue. I tried to throw the encrypt code into decrypt and change a few things but it was no good. |
| 24-April | COBOL | 2 | This is it. A day before its due and I’m struggling like crazy. I figured out how to do decrypt by just copying down the encrypt function, but I refuse to let it be because I don’t have the faintest idea how I can get the solve function to work. Up until now, the solve function generally calls decrypt which calls encrypt in some kind of loop or map. That won’t work if decrypt is functioning the way it is right now. Which will make it that much harder for solve to work. Thankfully, Professor Krog cancelled my Thursday Discrete classes so there is some last minute work I can do here. |
| 25-April | COBOL | 5 | I’m very opposed to doing work so close to a deadline. That’s Communication major stuff. But I was left without a choice sadly. One hour before this is due and I’ve only just figured it out. Despite hours to head banging and typing, I figured out how to do nested PERFORM loops properly. Its all because of these stupid periods and error messages that COBOL throws. It is a very silly and confusing syntax. Worse than ML using semicolons and places where it isn’t needed. The key came when I figured out how to execute a function call to the encrypt function in a PERFORM loop simultaneously resetting the data in the loop. Turns out, that if you don’t reset the variables (such as the length and the string) to 0 or a blank space every time you call encrypt, it fails. It will just spit out what you previously had in. This didn’t occur to me until I noticed SOLVE was spitting out powers of 2 when solving for HAL. It was doing every other solve of HAL.  Never again. COBOL and Erlang. As long as I live. |
| **TOTAL HOURS SPENT: ~53**  **Not counting time spent being haunted by COBOL & Erlang in my sleep.** | | | |