Roma Razdan
I pledge my honor that I have abided by the Stevens Honor System. Roma Razdan

## My Approach

While a little challenging, this project allowed me to grow my skills in arm assembly as well as work to correctly implement a code that performs the taylor series.

For the main method it was primitive I load the address of i into a register. As this value was an index, I had to load it into another parameter register. Moving this finally to a temporary register allowed me to prevent the value from changing. This concept in addition to subtracting 1 was used to index, since we are counting up to i - 1. Then I loaded x into d1, the double register, making sure d1 was working properly (which was one of the core registers). Additionally creating a count variable set to 1 allowed me to work seamlessly with the next loop where I would work through the taylor series. An additional variable set at 1 would also work for float conversion in attempts of taylor estimation. At the end of my main method I had to include the condition that if i = 0, there are no additional terms that must be included in the final result, then the program can go to done. Else, we utilize a double value for calculating the numerators of the remaining terms of the series and branch to the loop where this calculation is done (taylor loop (L1)).

In loop one, d1 is set to x and d3 is set to 1, thus x is the first iteration and x^2 is the second iteration and we continue these iterations until the end of this loop. The rest of this method entails calculating the denominator which is in terms of double as seen through the initialization of a variable to 1.

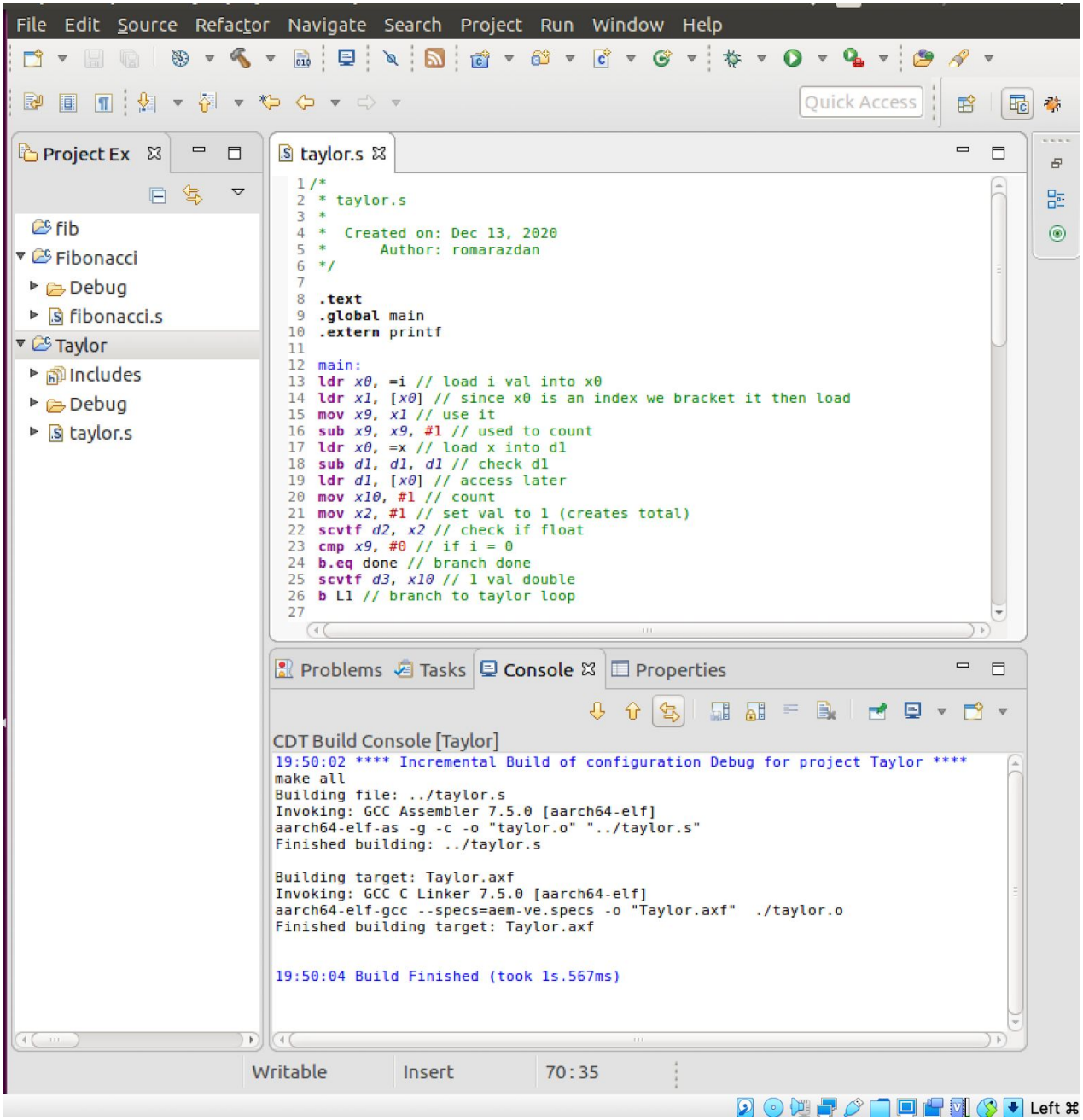
For factorial and loop 2 it was vital to calculate the factorial of the count using the same idea for loop one where conversions to double occurred. By branching back to the caller, the loop ran more seamlessly since branching would not occur in every case (less breaks). As recursion was not necessary nor the use of stack, I found branching in these cases to be the best option. If this case is not used, the loop runs until it reaches the count. By checking if count is at i - 1 we can say we are done with this method. Else, increment the count and repeat loop.

Finally, we check if d0 is 0 and then add the current estimation to d0. By loading fstring into x0 allowing d0 to print, we get the result of our taylor series and can exit the code in its entirety.

Overall, I found this project to be a great experience as I choose to utilize branching more so than recursion and stack as opposed to the first project. I also learned how double registers work and how to seamlessly run loops within other methods. Also I learned how to use mul (integer multiplication), fmul (double multiplication), fdiv (double division), and scvtf (conversion int to double). Lastly, I found it interesting to separate the numerator and denominator and then bring them together at the end of the code to produce the correct estimation!

Roma Razdan
I pledge my honor that I have abided by the Stevens Honor System. Roma Razdan

Screenshots:
Build Mode →



Roma Razdan

I pledge my honor that I have abided by the Stevens Honor System. Roma Razdan

Debug Mode →