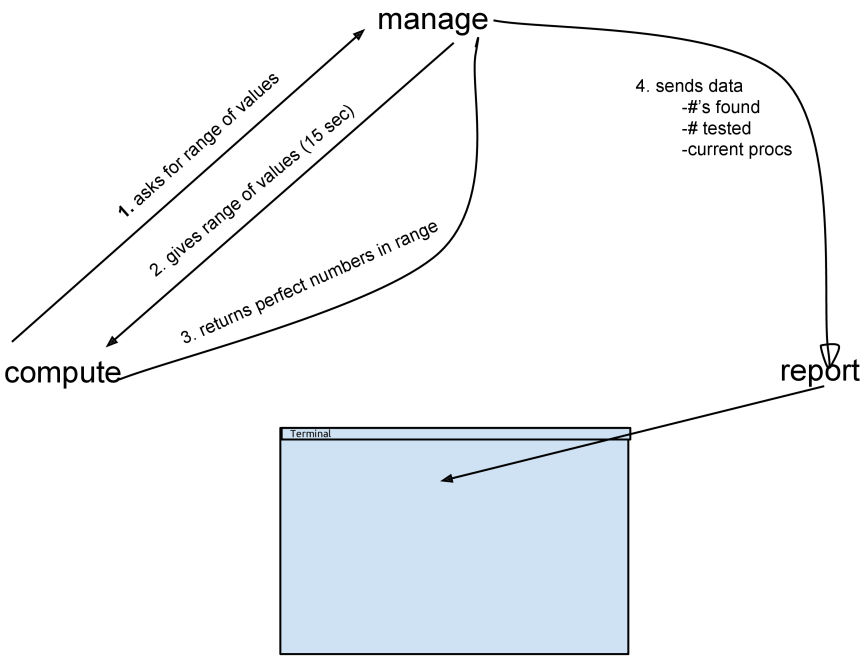
Program Design



Deviations From My Design

For the first time in this entire class, I actually did not have to deviate from my program design whatsoever in this assignment. I took a lot of time to draw out exactly what I needed to do, and it helped me out immensely whenever I got stuck during programming. I had to add on some functionality that I did not originally intend, however the actual program works exactly how I drew it out on paper and in Google Docs(see above).

This assignment has taught me that designing a flow chart for difficult programs is incredibly useful, and I intend on doing it whenever I embark on any medium to large sized program.

Problems Faced

- Figuring out how to parse XML and send it in between processes and languages was difficult.
- The hardest part of the assignment for me was figuring out how to change my basic client server architecture into a multiplexed one using select(). It took me about 5 hours.
- getopt.getopt in Python was slightly different than getopt() in C, so I had to figure out how to use it
- Another huge issue I had was figuring out how I wanted to deal with the -k signals from report going to the other programs.

Work Log

December 4, 2013 - Implemented basic client server architecture with Python, and wrote a compute algorithm in compute.c.

December 5, 2013 - Added basic XML parsing. Will be expanded upon later.

December 6, 2013 - Added client architecture in compute.c

December 7, 2013 - Expanded on XML parsing, and added it into compute.c. Also added a timing loop. compute.c and the server now pass performance characteristics, range of values to check, and computed perfect numbers.

December 8, 2013 - Added select() functionality in the server program. Still assuring it works correctly. Also added better formatted print statements to update the user on what is happening in real time.

December 8, 2013 - Finalized verbose print functions and ensured select() works well with the server and both clients all connected. Added signal handling