- Briefly describe the design of the program. How many processes and how many pipes are created with your solution? How does data flow from process to process?

  My solution requires four pipes and five processes. The program forks a new child for each command in the bash script. Every fork receives data from one pipe(except for the first which only writes to a pipe) and the process arguments then it passes its output to a pipe to the next process. The final process passes its output back to the parent and that is printed in the parent’s console.

- How did you test and debug your solution?

  I iteratively built the program and piped each new child’s output to the parent to confirm that the most recently added forked process was operating as expected. I tested my program using make test and observed the diff.

- When he was head of Bell Labs, Doug McIlroy summarized the "Unix philosophy" as follows:
  - Write programs that do one thing and do it well. Write programs to work together. Write programs to handle text streams, because that is a universal interface.

- How do pipes contribute to the Unix philosophy?

  Pipes allow simple programs which are good at one simple task to communicate with other programs that are also similarly only good at one task. With pipes, these simple programs can be assembled into more complex program structures.