1. Define the problem: State the problem you are trying to solve in clear and concise terms
2. Start with the end: Design what the user will see when they run your program
3. List the inputs: State the information and their data types needed to solve the problem
4. List the outputs: State the information and their data types the algorithm will produce as a result
5. Identify calculations: List any calculations or formulas needed to transform the inputs into outputs
6. List the steps: Use **pseudocode** or a **flowchart**to list the steps the computer will execute to solve the problem
7. Test the algorithm: Choose data sets and verify that your algorithm works

Problem: Need to get weighted average of time-to-next-input

End goal: Show a weighted scale of what commands result in the longest delay to next command

Inputs: Massive .csv list containing 100,000 inputs

Output: Sorted longest-to-shortest list with number of times instance occurred

Pseudocode: Push .csv into empty list; separate each entry into command, user, model, time; start bottom to top, list[x-1], get meta data, find next instance of user, get meta data, plot to list. Repeat until list end. Data compiled, run weighted calc on data and export to list.

Idea: Algorithm that takes any sentence and tells you which letter was most common. Good idea for a tie condition too!!!