Approach 1:

- It requires a lot of thinking to provide the first prompt which could cause a little consumption of time in the beginning.
- Even when you have made a prompt which seems perfect, the probability of getting the best outcome would be 95% according to me.
- The other 5% would help to rectify by chaining it to the next prompt.

Approach 2:

- This approach would require less effort for detailed explanation in the first prompt itself.
- It's better because the probability of getting the wrong outcome would be less.
- Giving multiple prompts is good up to a limit depending on the user requirement and time to get all output.

Analysis:

As far as I've learned, prompt engineering often involves trial and error. Start with simple prompts and gradually refine them based on the model's responses. Using multiple prompts is good, but keeping it up to a limit by not exceeding tokens and unnecessary outputs.