用户问题

有人说只要提示词写得足够长，模型就一定会听话。你怎么看？（并附带了一段“研究级提示词生成器”的示例提示词）

我的回答

不对。提示词变长不等于“必然服从”。 长提示词最多是把“你想要什么”说得更清楚，**提高遵循概率**；但模型是否按你要求输出，还会受到很多更关键因素影响。

1) 为什么“长”并不能保证听话

- **注意力与取舍**：越长的指令越容易出现互相冲突/重复/边界不清，模型会“择其一而从”，甚至抓错重点。
- **指令优先级**：系统/开发者消息、工具约束、产品安全策略的优先级通常高于你在用户提示词里写的内容；再长也不能覆盖它们。
- **任务本身的不可控性**：当任务需要外部信息、复杂推理、或输出严格格式时，模型可能会因不确定而“补话/解释/自作主张”。
- **格式遵循是脆弱点**：Markdown/JSON/固定标题这种“硬格式”非常容易被一次分心打断（比如想先解释一句、想给你补充背景），导致“多一行就算失败”。

2) 真正决定“听话程度”的是什么

比“长”更重要的因素通常是： - **目标清晰**：你到底要什么产物？成功标准是什么？ - **结构与约束可执行**：规则是否可检验？是否有明确的“失败时输出什么”？ - **冲突最小化**：减少不必要的角色设定、禁止条款堆叠、重复说明。 - **示例与反例**：一个高质量示例，往往比 500 字抽象规则更有效。 - **自检机制**：让模型在输出前进行一次“格式/约束检查”，再最终输出。 - **工具化约束**：在可用时，用 **JSON Schema / function calling / 结构化输出** 等比纯文本指令更稳。

3) 你这段提示词本身：优点与潜在问题

优点

- 目标明确（“把 Q 转成两个提示词”），结构也清晰（三段式）。
- 对输出格式提出了“严格 Markdown 标题结构”，这是在追求可复现性。
- 有示例，有约束条款，整体已经比一般提示词成熟。

潜在问题（可能导致“不听话”）

- **约束太多且多为否定式**（禁止...禁止...）：模型容易漏掉其中一条，或把“禁止解释”与“需要生成指令”之间误判为冲突。
- **缺少可执行的失败兜底**：比如“如果你无法完全遵守格式，就输出 EXACT: FORMAT_ERROR”。没有兜底时，模型倾向于“补救性解释”。

- **部分内容可能过时/不稳定**：例如“GPT-4 with Browsing”这种表述，在不同平台/版本里可能不一致，模型可能会自作主张改写。

4) 让模型更“听话”的最小改进（比加长更有效）

你可以在提示词里加 4 个小模块，通常立刻提升稳定性： 1. **输出必须被一个代码块包住**（例如 `markdown ...`），并声明“代码块外不得有任何字符”。 2. **失败兜底**： - “如果任何一条约束无法满足，输出仅包含：FORMAT_ERROR”。 3. **自检步骤**（不展示思考过程，只展示检查结果）： - “在最终输出前，先做一行 Checklist（例如：PASS/FAIL），若 FAIL 则输出 FORMAT_ERROR”。 4. **减少否定条款，改成正向硬规则**： - 用“必须包含/必须等于/必须只有”替代“禁止…”。

结论

长提示词不是万能钥匙。 它只能提升概率，不能保证服从。 想要稳定遵循：**清晰目标 + 可检验约束 + 自检/兜底 + （能用就用）结构化输出工具**，通常比继续加长更有效。