Double-click (or enter) to edit

2.PROMPT DESIGN AND EXPERIMENTATION

OBJECTIVE

- Create basic prompts for:8177
    - Text summarization
    - Code generation
    - Creative writing

  (a)BASIC PROMPT

TEXT SUMMARIZATION (prompt)

PROMPT

During his visit to the Galápagos Islands, Charles Darwin observed several species of finches that differed mainly in the shape and size of their beaks. He noted that finches on different islands had adapted to different food sources—some had long, narrow beaks for insects, while others had strong, thick beaks for cracking seeds. This variation helped Darwin conclude that species change over time through a process of natural selection, where traits that help organisms survive and reproduce become more common in future generations.

"Summarize the following passage in 3 sentences, focusing on how bird beaks helped Darwin develop his theory of evolution:" what worked,why it worked,what didn't worked

OUTPUT:

✍️ Example Summary Output: Darwin noticed that finches on the Galápagos Islands had different beak shapes suited to their food sources. These differences showed how birds had adapted to their environments over time. This led him to develop his theory of natural selection, where useful traits become common through survival and reproduction.

✅ What Worked:

- The summary captures the key link between beak shape, diet, and adaptation.
- It correctly connects this observation to Darwin’s theory of natural selection.
- The summary is concise, logical, and clearly structured in 3 sentences.

💡 Why It Worked:

- The prompt gave clear instructions, including both a focus (“how bird beaks helped Darwin...”) and a length limit.

- The input text was well-organized and directly connected observations to theory, which helped the summarizer focus on essential details.
- The structure of the prompt guided the model (or student) to retain cause and effect without adding extra context.

❌ What Didn’t Work:

- The summary did not include specific examples, like "long, narrow beaks for insects" or "thick beaks for seeds", which would strengthen clarity and depth.
- It may sound generic without naming actual species or island differences.
- Readers unfamiliar with natural selection might miss how beak variation directly proves Darwin's theory.
- Why This Happened:
    - he instruction didn't explicitly ask for examples or specific adaptations—it only said “summarize,” which can lead to generalizations.
    - AI models tend to prioritize general meaning over detail unless prompted otherwise.
- How to Improve the Prompt: You could revise it like this to get richer detail:

"Summarize the following passage in 3 sentences, including at least one example of how a bird's beak relates to its food source, and explain how this helped Darwin form his theory of natural selection."

CODE GENERATION

PROMPT

write a c program to make calendar of user suggesting year,day,month worked,why it worked,what didn't worked

```
#include <stdio.h>

void printMonthCalendar(int year, int month, int startDay) {
    int daysInMonth;

    // Determine number of days in month
    if (month == 2) {
        // Leap year check
        if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))
            daysInMonth = 29;
        else
            daysInMonth = 28;
    } else if (month == 4 || month == 6 || month == 9 || month == 11) {
        daysInMonth = 30;
    } else {
        daysInMonth = 31;
    }

    // Print header
    printf("\n\n     Calendar - %02d/%d\n", month, year);
    printf(" Su Mo Tu We Th Fr Sa\n");

    // Print initial spaces
    for (int i = 0; i < startDay; i++) {
        printf("   ");
    }

    // Print days
    for (int day = 1; day <= daysInMonth; day++) {
        printf("%3d", day);
        if ((day + startDay) % 7 == 0)
            printf("\n");
    }

    printf("\n");
}

int main() {
    int year, month, startDay;

    // Get user input
    printf("Enter year: ");
    scanf("%d", &year);

    printf("Enter month (1-12): ");
    scanf("%d", &month);

    printf("Enter starting day of the month (0=Sun, 1=Mon, ..., 6=Sat): ");
    scanf("%d", &startDay);

    // Display calendar
    printMonthCalendar(year, month, startDay);

    return 0;
}
```

- What Worked:
    - The program correctly prints the calendar layout for any given month and year.
    - It handles leap years accurately for February.
    - The weekday alignment and day counting are clear and readable.
- Why It Worked:
    - It uses correct logic for number of days per month.
    - The use of a loop to print days aligned under weekdays makes the structure consistent.

- What Didn’t Work:
    - The program requires the user to manually enter the starting weekday of the month, which most users won’t know.
    - It shows the month as a number (e.g., 06/2025) instead of a name (like June 2025), which is less user-friendly.
- Why It Didn’t Work:
    - There's no built-in logic to compute the weekday of the 1st of the month.
    - The program prioritizes basic structure over user experience or automation.

CREATIVE THINKING

PROMPT Write a short story where human organs can talk to each other worked,why it worked,what didn't worked OUTPUT :

🧠 “The Council Inside”

Every night, after Maya fell asleep, the organs gathered for their nightly meeting. It was their time to check in, argue, joke, and—mostly—complain.

Brain tapped his imaginary clipboard. “Alright team, let’s begin. Any updates?”

Stomach groaned. “Yeah. Can someone please explain the cheeseburger at midnight? I was winding down, and suddenly—bam! Grease, acid, chaos.”

Liver sighed. “And guess who had to clean up that mess? Me! I’ve been detoxing all day. No thanks to her third cup of coffee either.”

Heart, ever the emotional one, jumped in. “Can we focus on the bigger issue? She saw a sad dog video today and cried. I had to pump at double speed while she sniffled on the couch.”

Lungs coughed dramatically. “Meanwhile, I’m suffocating during her jogs. In... out... in... out... non-stop! I barely got a warning!”

Eyes blinked rapidly. “At least you don’t have to stare at a screen for eight hours. I’m drying out over here.”

Brain cleared his mental throat. “Okay, okay. Look, she’s doing her best. She just started drinking more water—”

Bladder cut in, clearly irritated. “Oh yes. Don’t think I didn’t notice! It’s been like a flood every hour. She must think I’m a public restroom.”

Legs, always too tired to argue, mumbled, “Can we just skip yoga tomorrow?”

Spine cracked. “No chance. She scheduled a stretching class. And if she slouches one more time, I’m leaving.”

Brain raised a neuron. “Everyone, calm down. Maya’s trying. She’s feeding us better, moving more, and sleeping longer. Let’s show a little unity, alright?”

The organs grumbled but nodded. After all, they were a team—different in form and function, but one in purpose.

As the alarm buzzed in the morning, they all braced themselves for another day in the life of a human.

✅ What Worked:

- Creative personification: Giving each organ a personality made the story fun and engaging.
- Humor and relatability: The organs complained about things real people experience—midnight snacks, crying at sad videos, too much screen time, etc.
- Clear structure: The meeting format helped organize the story logically, with each organ having a turn to speak.
- Subtle message: The story ends with a positive takeaway about health and teamwork inside the human body.

💡 Why It Worked:

- Readers can relate emotionally and physically to what the organs are going through.
- The story uses dialogue to keep the pace lively and avoid long exposition.
- The tone is light and humorous, making a biological concept enjoyable without being didactic.
- Each organ’s function was cleverly tied to their complaint, reinforcing the educational value.

❌ What Didn’t Work (or Could Be Improved):

- No strong plot: It’s more of a sketch or scene than a full story with conflict and resolution.
- Lack of depth: Some organs like kidneys, skin, or pancreas were left out, which could add more layers or variety.
- Repetitive tone: Since most organs were just complaining, the narrative tone felt a bit flat or one-note after a while.
- Short character development: The organs had traits, but not much evolution or surprise in their personalities. Moral: Even inside you, teamwork matters—so treat your body well. It’s listening (and complaining) more than you think. 😉

(b)Advanced technologies

- chain of thought prompting
- role prompting
- few shot prompting

OBJECTIVES

- CREATE 3 TYPES OF PROMPT BASED ON THESE COCEPTS:
    - Chain-of-thought prompting
    - Role prompting (e.g., "Act as a cybersecurity analyst...")
    - Few-shot promptin

1,CHAIN-OF-THOUGHTS PROMPTING

PROMPT

You are in a room with three doors.

- Door A leads to certain danger.
- Door B leads to a room where the floor collapses as soon as someone enters.
- Door C leads to a room with a deadly fire burning continuously. One of these doors is the safest option. Which door should you choose? worked,why it worked,what didn't worke

OUTPUT One of these doors is the safest option. Which door should you choose? Step-by-Step Solution:

Step 1: Analyze Each Door's Danger Level

- Door A: Leads to certain danger, but the exact threat is unknown.
- Door B: The floor collapses immediately upon entering, making survival impossible.
- Door C: The fire is burning continuously, meaning something must be keeping it going.

Step 2: Identify Possible Loopholes

- Since the fire in Door C has been burning for a long time, what would happen if it had been left alone?
- Fires require fuel to keep burning. If the room has been closed off, could the fire still have enough oxygen?

Step 3: Consider Real-World Knowledge

- A fire cannot burn forever without oxygen. If the room has been sealed, the fire may have burned out due to lack of air.
- Door A and Door B both have immediate dangers with no escape route.

Step 4: Analyze Door B

- Floor collapses as soon as someone enters" sounds dangerous. But — it doesn’t say the person dies or that there's a deadly trap below.

- Possibilities:

  The fall might not be fatal. The floor collapse could reveal a safe passage or another way out. Or, if you're careful, you might test or avoid stepping fully in. Conclusion: Potential to survive, depending on how you act.

Step 5: Make a logical choice

- Door A and C lead to guaranteed fatal outcomes.
- Door B involves uncertainty but no confirmation of death.
- Therefore, Door B is the safest logical option.

✅ Final Answer: Choose Door B Because it's the only door without certain, unavoidable death — it presents a chance, and in logic puzzles, uncertain danger is better than guaranteed death.

✅ What Worked:

- Elimination logic: The reasoning effectively ruled out Doors A and C by focusing on the certainty of danger.
  - Inference skills: It correctly inferred that Door B might not lead to immediate death, only risk — a key insight in logic-based riddles.
  - Structured thinking: The chain-of-thought approach made the reasoning clear and easy to follow, leading to a logical conclusion.

💡 Why It Worked:

- Because chain-of-thought prompting encourages deeper analysis, it avoids surface-level interpretations and helps identify subtle distinctions — like the difference between certain and potential danger.
- It mirrors how critical thinking works: assess all options, weigh risks, and choose the most viable path even under uncertainty.

❌ What Didn't Work:

- Assumptive logic: It assumed that falling through a collapsing floor (Door B) isn't fatal, without direct evidence.
- Lack of context clues: The solution relied on riddle-solving conventions (e.g., choosing the least obvious danger) which may not be intuitive for everyone.

No explanation of "why not A or C" beyond the surface: The analysis could've expanded slightly on why "certain danger" or "continuous fire" are logically more fatal.

ROLE PROMPTING

PROMPT

Pretend you are an astronaut who has traveled to Mars! Explain to me what the planet looks like and what it feels like to be there." IN ROLE PROMPTINGworked,why it worked,what didn't worke

🧑‍🚀 IN ROLE: ASTRONAUT ON MARS Log Entry: Sol 43 – Mars Mission Alpha

Wow. Mars is breathtaking — and eerie. The sky has a dusty orange hue, especially during sunset, where the light bends in strange, almost bluish tones near the horizon. The ground beneath my boots is dry, cracked, and reddish-brown everywhere — like walking across a giant rusted desert.

The silence is deafening. You can't hear a thing without your comms on — not even your own footsteps. Gravity here is only about one-third of Earth's, so each step feels bouncy, like you're walking on a trampoline in slow motion.

The atmosphere? Super thin and completely unbreathable — I rely on my suit 24/7. And the dust? It's everywhere. It clings to the suit, the visor, and our equipment. Yet despite all the challenges, standing here... looking out across an ancient, untouched world... it feels like being part of something historic. Something human.

🧠 ROLE PROMPTING – What Worked, What Didn't, and Why Immersive voice and perspective:
✅ What Worked:

- Immersive voice and perspective: By speaking as the astronaut, the explanation becomes personal and vivid.
- Sensory detail: Describing sight (orange sky, rusted terrain), sound (silence), and feel (light gravity) made the setting feel real.
- Emotional engagement: It conveyed awe, wonder, and isolation — feelings someone would have on Mars.
- Natural language: The astronaut didn't talk like a textbook — it sounded like a journal entry or conversation.

❌ What Didn't Work:

- Assumes listener background knowledge: It didn't explain what causes the low gravity or the orange sky — useful in a science context.
- Not much technical info: A teacher might want data: temperature, atmospheric pressure, etc.
- One-sided: Role prompting is powerful, but unless interactive, it doesn't check if the listener is following or has questions.

💡 Why It Worked:

- Role prompting helps the model (or student) embody a perspective, making the answer more authentic, emotionally rich, and imaginative.
- It allows you to blend factual accuracy with creative storytelling, which is perfect for engagement and memory retention.

FEW SHOT PROMPTING

PROMPT

Simple Few-Shot Prompt: Capitalization INPUT=orange.output=Orange ,give me the input and output of grapes,bsnana,pappaya,apple etc.worked,why it worked,what didn't worke

OUTPUT NPUT = orange OUTPUT = Orange

INPUT = grapes OUTPUT = Grapes

INPUT = bsnana OUTPUT = Bsnana

INPUT = pappaya OUTPUT = Pappaya

INPUT = apple OUTPUT = Apple

✅ What Worked:

- Clear Pattern Recognition
    - Each example consistently shows a lowercase input and a capitalized output.
    - This consistency helps models and learners easily infer the rule: capitalize the first letter.
- Simple and Focused Prompt
    - The prompt isolates just one skill: capitalization.
    - No distractions like extra formatting or mixed tasks.
- Useful Variety
    - Including multiple fruit names (grapes, banana, pappaya, apple) gives variety, reinforcing the pattern.
    - Even with a misspelling ("bsnana"), the model still learned to capitalize the first letter — showing it's focused on formatting, not spelling.

💡 Why It Worked:

- Few-shot prompting works best when examples are:
- Simple
- Repetitive
- Patterned clearly
- By providing multiple demonstrations, the model doesn't just guess — it learns the rule from context.
- It mirrors how humans learn: through seeing patterns applied over and over.

❌ What Didn't Work (or Could Be Improved):

- Spelling Error Left Untouched ("bsnana")
- While it shows the model's focus, it might confuse learners who think the spelling should also be corrected.

- If spelling correction is also expected, the prompt needs to clarify that as part of the task.
