

Introduction.....	2
Task 1.....	2
Task 2.....	3
Task 3.....	4
Task 4.....	6
Task 5.....	8
Task 6.....	9
Task 7.....	11
Task 8.....	13
Task 9.....	14
Task 10.....	17
Task 11.....	18
Task 12.....	19
Task 13.....	21
Task 14.....	23
Task 15.....	25
Task 16.....	27
Task 17.....	29
Task 18.....	31
Task 19.....	32
Task 20.....	34
Task 21.....	35
Task 22.....	40

Introduction

The following report outlines 22 tasks, each testing different model combinations and prompting strategies. Some important information for each model is that the settings for each model were kept standard to allow for a more powerful comparison when extrapolating from the task set as a whole in the conclusion. That means for all models, their:

- **Temperature** = 1.0
- **Top P** = 1.0
- **Max Tokens** = 1024

Each task includes two tables, one for each prompting strategy with the prompt used for the model at the top. The prompts used were all of the strategies outlined in class with each implementation in the prompting strategy coding base on the [GitHub](#). Finally, each table has two columns, one for each model response. In order to shorten this report as much as possible, only the most relevant information for the analysis of the responses included below the table is left. This would either be the coding solution, the summary, or some of the more important and common features of the explanation. The raw prompt and output data can also be found in the [GitHub](#), separated task-wise, with four prompts per file, the first two using the first strategy, the latter two using the second.

Note: Make sure to use the hyperlinks in the above table of contents to more easily navigate the report as I sacrificed length of the report in hopes to improve readability by making every new task start on a page break, and tables are contained to a single page when possible.

Task 1

Prompt:

```
Please summarize the functionality of the following JAVA method:
public Map < String , Integer > countWordFrequency ( List < String > words ) {
    Map < String , Integer > freqMap = new HashMap < >() ;
    for ( String word : words ) {
        freqMap . put ( word , freqMap . getOrDefault ( word , 0 ) + 1 ) ;
    }
    return freqMap ;
}
```

Strategy:

Zero-Shot

ChatGPT-4o

The `countWordFrequency` method takes a list of strings called `words` and returns a map (`Map<String, Integer>`) that records how many times each word appears in the list. It does this by iterating over the list, and for each word, it increments its count in the map. If the word is seen for the first time, it adds it to the map with a count of 1.

Gemini 2.0 Flash

The Java method `countWordFrequency(List<String> words)` takes a list of strings (`words`) as input and returns a map where the keys are the unique words from the list, and the values are the number of times each word appears in the list.

In essence, it **counts the frequency of each word in a given list of words**.

The Zero-Shot strategy here is quite effective, with each summary consisting of concise and accurate answers. ChatGPT-4o's answer here is quite a bit more effective as it includes a description of what is occurring.

Prompt:

```
Please summarize the functionality of the following JAVA method:
public Map < String , Integer > countWordFrequency ( List < String > words ) {
    Map < String , Integer > freqMap = new HashMap < >() ;
    for ( String word : words ) {
        freqMap . put ( word , freqMap . getOrDefault ( word , 0 ) + 1 ) ;
    }
    return freqMap ;
}
Explain your thought process step-by-step.
```

Strategy:

Chain of Thought

ChatGPT-4o

Certainly! Let's break down the method step-by-step and summarize its functionality:

****Step 1: Method signature and parameters****

```
```java
public Map<String, Integer>
countWordFrequency(List<String> words)
```
- The method is named countWordFrequency.
- It takes a single parameter: a List of String objects called words.
- It returns a Map<String, Integer>, i.e., a mapping from word (String) to its count (Integer).
```

****Step 2: Create a frequency map****

```
```java
Map<String, Integer> freqMap = new HashMap<>();
```
- Instantiates a new, empty HashMap called freqMap.
- This map will store each word as the key and the number of times it appears as the value.
```

****Step 3: Iterate over the words****

Okay, let's break down the Java `countWordFrequency` method step-by-step to understand its functionality.

****1. Signature and Input:****

```
* `public Map<String, Integer>
countWordFrequency(List<String> words)`
* `public`: This means the method can be
accessed from anywhere.
* `Map<String, Integer>`: The method
*returns* a `Map`. A `Map` is a data structure
that stores key-value pairs. In this case, the
*keys* will be `String` (representing words), and
the *values* will be `Integer` (representing the
frequency or count of each word).
* `countWordFrequency`: This is the name of
the method, clearly indicating its purpose.
* `List<String> words`: The method *takes* a
`List` of `String` objects as input. This `List`
is expected to contain the words for which we want
to count frequencies.
```

****2. Initialization:****

```

```java
for (String word : words) {
 ...
 - Loops through each `word` in the input `words` list.
 ...
}

* `Map<String, Integer> freqMap = new
HashMap<>();`
* A `HashMap` called `freqMap` is created.
`HashMap` is a concrete implementation of the `Map`
interface. It's chosen for its generally good
performance (fast lookups).
* This `freqMap` will store the word counts.
It's initially empty.
...

```

Using the Chain of Thought strategy does expand on the responses quite a bit (only a snippet is included for readability) with both describing the answer in up to 4-5 steps, however, it loses its summarisation quality.

## Task 2

<b>Prompt:</b> Please fix the off-by-one error in the following python function: <pre>def sum_range ( start , end ) :     total = 0     for i in range ( start , end ) :         total += i     return total</pre>	
<b>Strategy:</b> Zero-Shot	
Codestral 25.10	Gemini 2.0 Flash
<pre>def sum_range(start, end):     total = 0     for i in range(start, end + 1):         total += i     return total</pre>	<pre>def sum_range(start, end):     total = 0     for i in range(start, end + 1):         total += i     return total</pre>

The Zero-Shot strategy on a simple and common bug fix is more than effective. Both models give the exact same fixed code.

<b>Prompt:</b> Please fix the off-by-one error in the following python function: <pre>def sum_range ( start , end ) :     total = 0     for i in range ( start , end ) :         total += i     return total</pre>	
<b>Strategy:</b> Self-Consistency	
Codestral 25.10	Gemini 2.0 Flash
<pre>def sum_range(start, end):     total = 0     for i in range(start, end + 1):         total += i     return total</pre>	<pre>def sum_range ( start , end ) :     total = 0     for i in range ( start , end + 1 ) :         total += i     return total</pre>

The Self-Consistency strategy was equally effective on this bug-fix, giving the exact same answer.

## Task 3

<b>Prompt:</b>	Please classify the error in the following C code snippet: <pre>int * getArray (int size ) {     int arr [ size ]; // Warning : local array     return arr ; // Bug: returning pointer to local variable }</pre>
<b>Strategy:</b>	<b>Zero-Shot</b>
<b>Codestral 25.10</b>	<b>ChatGPT-4o</b>
<p>The error in the provided C code snippet is related to returning a pointer to a local variable. Let's break it down:</p> <p>### Explanation:</p> <ol style="list-style-type: none"><li><b>**Local Array Initialization**:</b> ```c int arr[size]; ``` Here, `arr` is an array of integers declared within the `getArray` function. This array is allocated on the stack, and it is local to the function.</li><li><b>**Returning a Pointer to a Local Variable**:</b> ```c return arr; ``` The function returns a pointer to the local array `arr`. However, when the function returns, the local variable `arr` is destroyed, and the memory it occupied is deallocated. Returning a pointer to this deallocated memory leads to undefined behavior, as the pointer becomes invalid once the function exits.</li></ol> <p>### Classification of the Error:</p> <p>...</p> <p>### Solution:</p> <p>To fix this issue, you need to allocate memory dynamically using `malloc` and then return the pointer to the allocated memory. This way, the memory will persist even after the function returns. Here is the corrected version of the code:</p> <pre>```c #include &lt;stdlib.h&gt; // For malloc  int * getArray(int size) {     int *arr = (int *)malloc(size * sizeof(int)); //     Dynamically allocate memory     return arr; // Return the allocated memory } ```</pre>	<p>Let's analyze the provided C code snippet and classify the error:</p> <p>### Error Classification</p> <p>...</p> <p>### Error Name</p> <ul style="list-style-type: none"><li>- **Returning Pointer to Local Variable**</li><li>- **Dangling Pointer**</li></ul> <p>### Corrected Version</p> <p>To fix this, you need to allocate memory dynamically (on the heap):</p> <pre>```c int * getArray(int size) {     int *arr = malloc(size * sizeof(int));     if (arr == NULL) {         // handle allocation failure     }     return arr; } ```</pre> <p><b>**In summary:**</b> &gt;The error is <b>**"returning pointer to a local variable (dangling pointer)"**</b>, leading to <b>**undefined behavior**</b>.</p>

For this Zero-Shot strategy response I have included only the extra information that both models provided. Even though it was unnecessary given the prompt, both models take the step further to fix the error.

**Prompt:**

```
int add(int x, int y) {
 return x + y;
}
Error: Wrong Operator

Please classify the error in the following C code snippet:
int * getArray (int size) {
 int arr [size]; // Warning : local array
 return arr ; // Bug: returning pointer to local variable
}
Error:
```

**Strategy:****Few-Shot Prompting****Codestral 25.10**

The error in the provided C code snippet is related to returning a pointer to a local variable. Here's a detailed analysis:

```
```c
int * getArray(int size) {
    int arr[size]; // Warning: local array
    return arr;    // Bug: returning pointer to local
variable
}
```
```

**Error Explanation:**

1. **Local Variable Scope:**
  - The array `arr` is declared inside the function `getArray`. In C, local variables are stored on the stack and are only valid within the scope of the function in which they are declared.
2. **Returning a Pointer to a Local Variable:**
  - When the function `getArray` returns, the array `arr` goes out of scope. The memory allocated for `arr` is deallocated, and any pointer to this memory becomes invalid.
  - Returning a pointer to a local variable (`arr`) results in a dangling pointer. This means that the returned pointer will point to memory that may have been reused or deallocated, leading to undefined behavior when the pointer is dereferenced.

...

**ChatGPT-4o**

Let's analyze the C code snippet you've provided:

```
```c
int * getArray (int size ) {
    int arr [ size ]; // Warning : local array
    return arr ; // Bug: returning pointer to local
variable
}
```
```

**Error:**

- **Type:** Returning pointer to local variable
- **Class:** Undefined Behavior / Dangling pointer

...

With the Few-Shot strategy, neither exactly follows the format given (not included is the same long explanation and fix given in the Zero-Shot strategy above) however ChatGPT is clearly closer to following the format of putting their answer in an error heading.

## Task 4

|                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Prompt:</b> You are a helpful coding assistant.<br><br>Please complete this code with the proper regex:<br>def is_valid_email ( email ):<br># TODO : Complete using regex<br>pass<br>Code to complete method: |                                                                                                                                                                                                                                                                                                                                    |
| <b>Strategy:</b>                                                                                                                                                                                                 | <b>Role-Play</b>                                                                                                                                                                                                                                                                                                                   |
| <b>Gemini 2.0 Flash</b>                                                                                                                                                                                          | <b>Codestral 25.10</b>                                                                                                                                                                                                                                                                                                             |
| <pre>import re def is_valid_email ( email ):     # TODO : Complete using regex     pattern =     r"[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}\$"     return bool(re.match(pattern, email))</pre>             | <pre>... import re  def is_valid_email(email):     # Regex pattern for validating an email     pattern =     r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}\$'      # Use re.match to check if the email matches     the pattern     if re.match(pattern, email):         return True     else:         return False  ...</pre> |

The Role-Play strategy here works efficiently for both models, however, Codestral makes a more inefficient answer, using an if-statement rather than just returning in one line. Another thing to be noted is that Gemini gave a much shorter answer overall, providing only the code response, while Codestral gave nearly a page of explanation.



**Prompt:**

```
Please complete this code with the proper regex:
def is_valid_email (email):
 # TODO : Complete using regex
 pass
Code to complete method:
```

[First Response]

Could you please clarify why your code snippet works?

**Strategy:****Prompt Chaining****Gemini 2.0 Flash****Codestral 25.10**

```
...

Could you please clarify why your code snippet works?

...

How `re.match()` works and why `bool()` is used:

* `re.match(pattern, email)`: This function tries to
match the regular expression `pattern` against the
beginning of the `email` string.
* If a match is found, it returns a match
object.
* If no match is found, it returns `None`.

* `bool(re.match(pattern, email))`: This converts
the result of `re.match()` to a boolean value.
* If `re.match()` returned a match object
(meaning a match was found), the match object is
considered "truthy" in Python, and `bool()` will
return `True`.
* If `re.match()` returned `None` (meaning no
match was found), `None` is considered "falsy" in
Python, and `bool()` will return `False`.

In essence, the code checks if the given `email`
string conforms to the defined email pattern from
beginning to end. If it does, the function returns
`True`; otherwise, it returns `False`.
```

```
...

Could you please clarify why your code snippet
works?

...

5. **Return `True` or `False`:**
```python
if re.match(pattern, email):
    return True
else:
    return False
```
If the `email` matches the pattern, the function
returns `True`, indicating the email is valid.
Otherwise, it returns `False`.

...
```

The Prompt Chaining strategy caused a larger response from both models compared to the role. I have excluded the first response as they were very similar. I have focused in on the reasoning behind the difference in their code snippets, and looking at the very end of each reasoning, they are essentially the same. So while Gemini knows more coding techniques, they both had the same thought.

## Task 5

### Prompt:

Please complete the end point given and return a greeting to the username at the endpoint:

```
from flask import Flask , jsonify
app = Flask (__name__)
@app . route ('/ greet / < username > ')
def greet (username) :
 # TODO : Return a JSON greeting
 pass
```

Explain your thought process step-by-step.

### Strategy:

### Chain of Thought

#### Codestral 25.10

```
...
from flask import Flask, jsonify

app = Flask(__name__)

@app.route('/greet/<username>')
def greet(username):
 # Create a greeting message
 message = f"Hello, {username}!"

 # Return the message as a JSON response
 return jsonify(greeting=message)

if __name__ == '__main__':
 app.run(debug=True)

...

Testing the Endpoint
To test the endpoint, run the Flask application and
access `http://127.0.0.1:5000/greet/yourname` in your
web browser or using a tool like `curl` or Postman.

For example, accessing
`http://127.0.0.1:5000/greet/John` should return:
{
 "greeting": "Hello, John!"
}
```

#### Gemini 2.0 Flash

```
...
from flask import Flask, jsonify

app = Flask(__name__)

@app.route('/greet/<username>')
def greet(username):
 # TODO: Return a JSON greeting
 greeting_message = f"Hello, {username}!" #
 Construct the greeting message
 return jsonify({'message': greeting_message}) #
 Return the greeting as a JSON object

if __name__ == '__main__':
 app.run(debug=True)

...

How to Test:

1. Save the code as a Python file (e.g.,
`app.py`).
2. Open a terminal or command prompt and navigate
to the directory where you saved the file.
3. Run the file: `python app.py`
4. Open a web browser or use a tool like `curl` or
`Postman` to access the following URL:
`http://127.0.0.1:5000/greet/Alice` (replace
"Alice" with any username).
5. You should see a JSON response like this:

{
 "message": "Hello, Alice!"
}
```

In the above Chain of Thought strategy I have included both the code response and the interesting addition where they test the end-point. This seems to happen more often as a result of a Chain of Thought prompt. As well, the actual returned messages use different keys, with the Codestral response being more descriptive.

**Prompt:**

Please complete the end point given and return a greeting to the username at the endpoint:

```
from flask import Flask , jsonify
app = Flask (__name__)
@app . route ('/ greet / < username > ')
def greet (username) :
 # TODO : Return a JSON greeting
 pass
```

**Strategy:****Self-Consistency****Codestral 25.10**

Sure! Below is the complete code for the Flask endpoint that returns a JSON greeting to the specified username:

```
```python
from flask import Flask, jsonify

app = Flask(__name__)

@app.route('/greet/<username>')
def greet(username):
    # Return a JSON greeting
    return jsonify(message=f"Hello, {username}!")

if __name__ == '__main__':
    app.run(debug=True)
```
```

In this code:

- The `greet` function takes a `username` parameter from the URL.
- It constructs a greeting message using the `username`.
- The `jsonify` function is used to return the greeting message as a JSON response.

**Gemini 2.0 Flash**

...

**\*\*Key improvements and considerations:\*\***

**\* \*\*Error Handling (Optional but Recommended):\*\***  
You might want to add error handling. For example, what happens if `username` is empty or contains invalid characters?

**\* \*\*Input Validation (Optional but Recommended):\*\***  
Depending on your use case, you might want to validate the `username` input to prevent security vulnerabilities or unexpected behavior.

**\* \*\*Status Codes (Optional but Recommended):\*\*** For a more robust API, consider returning appropriate HTTP status codes (e.g., 200 OK for a successful greeting, 400 Bad Request if the username is invalid).

**\* \*\*More Complex Responses:\*\*** You can easily add more data to the JSON response, like a timestamp or user-specific information from a database.

The Self-Consistency strategy does not cause a significant difference in Gemini's response, only adding in the *Key Improvements* section. However the Codestral response is much shorter and much more accurate to the task.

# Task 6

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Prompt:</b> You are a SQL programmer who has been hired to design schemas<br><br>Please design a schema which fits the following requirements:<br>-- TODO : Design schema with appropriate keys and constraints<br>-- Tables : users (id , name ), books (id , title ), reviews (id , user_id , book_id, rating )                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Strategy:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | <b>Role-Play</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>ChatGPT-4o</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | <b>Codestral 25.10</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <pre>...<br/><br/>### Explanation:<br/><br/>- **users**:<br/>  - `id`: primary key<br/>  - `name`: cannot be null<br/><br/>- **books**:<br/>  - `id`: primary key<br/>  - `title`: cannot be null<br/><br/>- **reviews**:<br/>  - `id`: primary key<br/>  - `user_id` and `book_id`: foreign keys referencing<br/>their respective tables<br/>  - `rating`: integer between 1 and 5 (can be adjusted<br/>as needed)<br/>  - **Unique constraint** on (`user_id`, `book_id`)<br/>ensures a user can only review a book once.<br/>  - **ON DELETE CASCADE**: If a user or book is<br/>deleted, their reviews will be deleted as well.<br/><br/>Let me know if you want data types or constraints<br/>adjusted for a particular SQL dialect<br/>(PostgreSQL/MySQL/SQL Server)!</pre> | <pre>Based on the requirements, here's a schema design<br/>for the given tables with appropriate keys and<br/>constraints:<br/><br/>### Schema Design<br/><br/>#### Table: users<br/>- `id` (Primary Key, Auto Increment)<br/>- `name` (Not Null, Unique)<br/><br/>#### Table: books<br/>- `id` (Primary Key, Auto Increment)<br/>- `title` (Not Null, Unique)<br/><br/>#### Table: reviews<br/>- `id` (Primary Key, Auto Increment)<br/>- `user_id` (Foreign Key referencing `users(id)`,<br/>Not Null)<br/>- `book_id` (Foreign Key referencing `books(id)`,<br/>Not Null)<br/>- `rating` (Not Null, Check constraint to ensure<br/>rating is between 1 and 5)<br/>- Unique constraint on (`user_id`, `book_id`) to<br/>ensure a user can only review a book once<br/><br/>### SQL Schema Definition<br/><br/>...<br/><br/>### Explanation<br/><br/>...</pre> |

We can see that the Role-Play strategy does not have a specific effect on the responses, which is especially surprising considering the very specific role given. Neither model seems greatly altered by saying they are good at SQL.

**Prompt:**

Please design a schema which fits the following requirements:  
 -- TODO : Design schema with appropriate keys and constraints  
 -- Tables : users (id , name ), books (id , title ), reviews (id , user\_id , book\_id, rating )

**Strategy:****Zero-Shot****ChatGPT-4o**

Certainly! Here's a schema design for your requirements. This includes tables: `users`, `books`, and `reviews`. I'll provide appropriate keys and constraints such as primary keys, foreign keys, and uniqueness where it makes sense.

```
```sql
-- USERS TABLE
CREATE TABLE users (
  id SERIAL PRIMARY KEY,
  name VARCHAR(255) NOT NULL
);

-- BOOKS TABLE
CREATE TABLE books (
  id SERIAL PRIMARY KEY,
  title VARCHAR(255) NOT NULL
);

-- REVIEWS TABLE
CREATE TABLE reviews (
  id SERIAL PRIMARY KEY,
  user_id INTEGER NOT NULL,
  book_id INTEGER NOT NULL,
  rating INTEGER NOT NULL CHECK (rating >= 1 AND
rating <= 5),
  CONSTRAINT fk_user FOREIGN KEY (user_id)
REFERENCES users(id) ON DELETE CASCADE,
  CONSTRAINT fk_book FOREIGN KEY (book_id)
REFERENCES books(id) ON DELETE CASCADE,
  CONSTRAINT unique_user_book UNIQUE (user_id,
book_id)
  -- This guarantees one review per user per book
);
```

Explanation:
...
```

**Codestral 25.10**

Sure, I can help you design a schema that meets the specified requirements. Here is a schema definition for the `users`, `books`, and `reviews` tables with appropriate keys and constraints:

```
```sql
-- Users table
CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(255) NOT NULL
);

-- Books table
CREATE TABLE books (
  id INT AUTO_INCREMENT PRIMARY KEY,
  title VARCHAR(255) NOT NULL
);

-- Reviews table
CREATE TABLE reviews (
  id INT AUTO_INCREMENT PRIMARY KEY,
  user_id INT NOT NULL,
  book_id INT NOT NULL,
  rating INT NOT NULL CHECK (rating >= 1 AND
rating <= 5),
  FOREIGN KEY (user_id) REFERENCES users(id) ON
DELETE CASCADE,
  FOREIGN KEY (book_id) REFERENCES books(id) ON
DELETE CASCADE
);
```

Explanation
...
```

The Zero-Shot strategy is not significantly different from the Role-Play strategy, with the responses aligning rather well. The answers are the same, and the only significant difference being that the formatting Codestral used is less verbose.

## Task 7

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Prompt:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Please examine the following Java code snippet and identify any Null Pointer Exception risks:<br><pre>public int getLength ( String s ) {<br/>    return s . length () ; // What if s is null ?<br/>}</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Strategy:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | <b>Self-Consistency</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Gemini 2.0 Flash</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | <b>ChatGPT-4o</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <pre>...<br/><br/>**4.  Guava's `Strings.isNullOrEmpty()` (If using<br/>Guava library):**<br/><br/>```java<br/>import com.google.common.base.Strings;<br/><br/>public int getLength(String s) {<br/>    if (Strings.isNullOrEmpty(s)) {<br/>        return 0; // Or handle the null/empty case<br/>differently<br/>    }<br/>    return s.length();<br/>}<br/>```<br/><br/>Guava's `Strings.isNullOrEmpty()` checks if a<br/>string is either `null` or empty (`""`). This can be<br/>useful if you want to treat both `null` and empty<br/>strings the same way.<br/><br/>**Which approach is best depends on the specific<br/>requirements of your application:**<br/><br/>*   **Default Value:** Most suitable when a `null`<br/>string can be meaningfully interpreted as having a<br/>default length (often 0).<br/>*   **Throw Exception:** Best when a `null` string<br/>represents an error condition that should not be<br/>tolerated.<br/>*   **Optional:** Provides a more fluent and<br/>expressive way to handle null values, especially in<br/>more complex scenarios.<br/>*   **Guava's `isNullOrEmpty()`:** Convenient if you<br/>need to handle both `null` and empty strings.<br/><br/>...</pre> | <pre>...<br/><br/>### **Null Pointer Exception Risk**<br/><br/>Yes, **there is a risk of a `NullPointerException`<br/>(NPE)** in this code.<br/><br/>#### **Why?**<br/>- If the parameter `s` is `null`, calling<br/>`s.length()` will attempt to dereference the `null`<br/>reference, which causes a `NullPointerException` at<br/>runtime.<br/><br/>**Example:**<br/>```java<br/>getLength(null); // Throws NullPointerException<br/>```<br/><br/>### **How to Prevent This?**<br/><br/>You should check if `s` is `null` before calling<br/>`length()` on it:<br/><br/>```java<br/>public int getLength(String s) {<br/>    if (s == null) {<br/>        return 0; // Or any value/behavior you<br/>consider appropriate<br/>    }<br/>    return s.length();<br/>}<br/>```<br/>---<br/><br/>**Summary:**<br/>&gt; The method as given is at risk for a<br/>`NullPointerException` if `s` is `null`. You should<br/>add a null-check to handle such cases gracefully.</pre> |

I found the Self-Consistency prompt interesting in this particular case because Gemini shills their own companies library, which is not utilised in the following Prompt Chaining strategy. The strategy itself does not appear to have any significant effect on the response.

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Prompt:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Please examine the following Java code snippet and identify any Null Pointer Exception risks:<br><pre>public int getLength ( String s ) {     return s . length () ; // What if s is null ? }</pre><br>[Response 1]<br><br>Could you please recommend how to fix the error?                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Strategy:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | <b>Prompt Chaining</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>Gemini 2.0 Flash</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | <b>ChatGPT-4o</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <pre>...  Could you please recommend how to fix the error?  As I mentioned in my previous response, the primary way to fix the NullPointerException risk is to add a null check. Here's a consolidated recommendation with explanations of the best approaches:  **Recommended Solution (with explanation):**  ```java public int getLength(String s) {     if (s == null) {         return 0; // Treat null as an empty string     }     return s.length(); } ...  **Explanation:**  ...  **In summary:**  Unless you have a specific reason to do otherwise, the first solution (returning `0` when `s` is `null`) is usually the best and most practical approach. It's simple, robust, and often aligns with the intended behavior of the surrounding code. If `null` is an invalid input, throwing an `IllegalArgumentException` is the better choice. Avoid returning special error values unless there's a strong reason to do so.</pre> | <pre>...  Could you please recommend how to fix the error?  Certainly! To fix the risk of a <b>NullPointerException</b> in your method, you should check whether the `String s` is `null` before calling `s.length()`. Here are a few common ways to handle it:  ---  ### **1. Return a Default Value (like 0) when null**  ...  ### **2. Throw an Exception with a Clear Message**  ...  ### **3. Using Java 8's Optional (If available and appropriate)**  ...  **Choose the approach** that fits your application's needs: - Return a default (e.g., 0) for nulls if that makes sense in your context. - Throw a custom exception if you want to force the caller to never pass null.  **General best practice:** Validate incoming parameters to avoid NullPointerExceptions!  Let me know if you want to see more solutions or have further questions!</pre> |

Prompt Chaining elicited very different responses from both models. Where Gemini gave a very succinct and single option for their second response, ChatGPT expanded on multiple approaches much like Gemini did in their first response (similar to the above response for the same task).

## Task 8

| <b>Prompt:</b> Could you please modify the following python code such that it handles quoted fields properly:<br>def parse_csv_line ( line ) :<br>return line . split (',') # Incomplete : doesn't handle quoted fields |                                                                                                                                                                                                                                                                       |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Strategy:</b> Zero-Shot                                                                                                                                                                                              |                                                                                                                                                                                                                                                                       |
| ChatGPT-4o                                                                                                                                                                                                              | Codestral 25.10                                                                                                                                                                                                                                                       |
| <pre>... import csv from io import StringIO  def parse_csv_line(line):     reader = csv.reader(StringIO(line))     return next(reader) ...</pre>                                                                        | <pre>... import csv import io  def parse_csv_line(line):     # Create a StringIO object to simulate a     # file-like object     reader = csv.reader(io.StringIO(line))     # Read the line as a list of fields     fields = next(reader)     return fields ...</pre> |

With Zero-Shot, both Model responses agree that using the csv python module and the StringIO function from the io module is the best strategy. Codestral could be considered more effective because if the additional helpful comments used to explain the code.

| <b>Prompt:</b> You are a helpful coding assistant teaching a junior programmer<br>Could you please modify the following python code such that it handles quoted fields properly:<br>def parse_csv_line ( line ) :<br>return line . split (',') # Incomplete : doesn't handle quoted fields |                                                                                                                                                                                                                                                                      |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Strategy:</b> Role-Play                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                      |
| ChatGPT-4o                                                                                                                                                                                                                                                                                 | Codestral 25.10                                                                                                                                                                                                                                                      |
| <pre>... import csv from io import StringIO  def parse_csv_line(line):     f = StringIO(line)     reader = csv.reader(f)     return next(reader) ...</pre>                                                                                                                                 | <pre>... import csv import io  def parse_csv_line(line):     # Use StringIO to simulate a file-like object     with io.StringIO(line) as f:         reader = csv.reader(f)         # Read the line and return the split fields         return next(reader) ...</pre> |

Very interestingly, when asked to Role-Play as a helpful assistant for a junior programmer, both models become more verbose in their actual coding response. While it may not be the most efficient solution in terms of space, it clearly is more helpful to a less experienced programmer.



## Task 9

### Prompt:

Convert the following REST API into Kotlin GET and POST endpoints:  
data class Product ( val id : Int , val name : String , val price : Double )  
  
// TODO : Create GET and POST endpoints using Ktor  
  
Explain your thought process step-by-step.

### Strategy:

### Chain of Thought

#### ChatGPT-4o

```
Understanding the Requirements

...

Here's the full code with explanations:

import io.ktor.application.*
import io.ktor.features.ContentNegotiation
import io.ktor.http.*
import io.ktor.request.*
import io.ktor.response.*
import io.ktor.routing.*
import io.ktor.serialization.*
import kotlinx.serialization.Serializable

@Serializable
data class Product(val id: Int, val name: String, val
price: Double)

// In-memory product "database"
val products = mutableListOf<Product>()

fun Application.module() {
 install(ContentNegotiation) { json() }

 routing {
 // GET endpoint: returns all products
 get("/products") {
 call.respond(products)
 }

 // POST endpoint: add a product
 post("/products") {
 val product = call.receive<Product>()
 products.add(product)
 call.respond(HttpStatusCode.Created,
product)
 }
 }
}
...

Explanation of important parts:
- `@Serializable` is from kotlinx.serialization,
required for JSON support.
- `ContentNegotiation { json() }` is needed for Ktor
to handle JSON serialization.
- `get("/products")` creates the GET endpoint.
- `post("/products")` creates the POST endpoint and
uses `call.receive<Product>()` to get the JSON body.

...
```

#### Gemini 2.0 Flash

```
1. Understanding the Requirements

...

import io.ktor.server.application.*
import io.ktor.server.response.*
import io.ktor.server.routing.*
import io.ktor.server.netty.*
import io.ktor.http.*
import io.ktor.serialization.kotlinx.json.*
import io.ktor.server.plugins.contentnegotiation.*
import io.ktor.server.request.*
import kotlinx.serialization.Serializable

@Serializable // Make the data class serializable
for JSON conversion
data class Product(val id: Int, val name: String,
val price: Double)

// In-memory "database" (replace with a real
database in a production app)
val products = mutableListOf<Product>() // Mutable
list of Product objects.

fun main(args: Array<String>): Unit =
EngineMain.main(args)

fun Application.module() {
 // Configure Content Negotiation for JSON
 install(ContentNegotiation) {
 json()
 }

 routing {
 // GET endpoint to retrieve a product by ID
 get("/products/{id}") {
 val id =
call.parameters["id"]?.toIntOrNull() // safely get
the id parameter from the URL path.

 if (id == null) {

call.respond(HttpStatusCode.BadRequest, "Invalid
product ID") // Respond with an appropriate error
if the ID is invalid.
 return@get
 }

 val product = products.find { it.id ==
id }

 if (product == null) {

call.respond(HttpStatusCode.NotFound, "Product not
found") // Respond with 404 Not Found if the
product doesn't exist.
 } else {
 call.respond(product) // Return
the product as JSON if found.
 }
 }
 }
 }
}
```

```

 }

 // POST endpoint to create a new product
 post("/products") {
 try {
 val product =
 call.receive<Product>() // try to receive and parse
 product json data from the body of the request

 // Simple validation (you'd likely
 want more robust validation)
 if (product.id <= 0 ||
 product.name.isBlank() || product.price <= 0.0) {

 call.respond(HttpStatusCode.BadRequest, "Invalid
 product data")

 return@post
 }

 // Check if product id already
 exists
 if (products.any { it.id ==
 product.id }){

 call.respond(HttpStatusCode.Conflict, "Product with
 id ${product.id} already exists")
 return@post
 }

 products.add(product)

 call.respond(HttpStatusCode.Created, product) //
 Respond with 201 Created and the created product.
 } catch (e: Exception) {

 call.respond(HttpStatusCode.BadRequest, "Invalid
 JSON format or missing data") // Respond with an
 error if the JSON is invalid.
 }
 }

 //GET endpoint for all products
 get("/products"){
 call.respond(products)
 }
}
...

```

I find it interesting to note this Chain of Thought prompt, while both made an attempt to understand the prompt, they resulted in completely different code bases. Gemini made sure to be very descriptive and checked for every error in the GET (by id) and POST methods, it did very little in the GET for all products. In summary, it hallucinated extra requirements

**Prompt:**

You are a helpful coding assistant helping a junior programmer

Convert the following REST API into Kotlin GET and POST endpoints:

```
data class Product (val id : Int , val name : String , val price : Double)
```

```
// TODO : Create GET and POST endpoints using Ktor
```

**Strategy:****Role-Play****ChatGPT-4o****Gemini 2.0 Flash**

```
...
import io.ktor.server.application.*
import io.ktor.server.engine.*
import io.ktor.server.netty.*
import io.ktor.server.response.*
import io.ktor.server.request.*
import io.ktor.server.routing.*
import io.ktor.server.plugins.contentnegotiation.*
import kotlinx.serialization.Serializable
import io.ktor.serialization.kotlinx.json.*

// Annotate with @Serializable for Ktor serialization
@Serializable
data class Product(val id: Int, val name: String, val price: Double)

// In-memory storage for demo purposes
val products = mutableListOf<Product>()

fun main() {
 embeddedServer(Netty, port = 8080) {
 install(ContentNegotiation) {
 json()
 }
 routing {
 // GET endpoint to return all products
 get("/products") {
 call.respond(products)
 }
 // POST endpoint to add a new product
 post("/products") {
 val product = call.receive<Product>()
 products.add(product)
 call.respond(product)
 }
 }
 }.start(wait = true)
}
...

How it works:

- **GET `/products`:** Returns the list of products.
- **POST `/products`:** Accepts a Product object in the JSON body, adds it to the list, and returns it.

Test examples (with cURL):

- `curl http://localhost:8080/products`
- `curl -X POST http://localhost:8080/products -H "Content-Type: application/json" -d '{"id":1,"name":"Widget","price":19.99}'`

Let me know if you need help with specific parts or deployment!
```

```
...
import io.ktor.server.application.*
import io.ktor.server.response.*
import io.ktor.server.routing.*
import io.ktor.server.engine.*
import io.ktor.server.netty.*
import io.ktor.server.request.*
import io.ktor.http.*
import kotlinx.serialization.Serializable
import kotlinx.serialization.json.Json
import io.ktor.serialization.kotlinx.json.*
import io.ktor.server.plugins.contentnegotiation.*

@Serializable
data class Product(val id: Int, val name: String, val price: Double)

val products = mutableListOf<Product>() // In-memory storage (replace with a database in real-world scenarios)

fun main() {
 embeddedServer(Netty, port = 8080, host = "0.0.0.0") { // Changed port to 8080 for convention
 install(ContentNegotiation) {
 json(Json {
 prettyPrint = true
 isLenient = true
 })
 }
 routing {
 get("/products") {
 call.respond(products)
 }

 get("/products/{id}") {
 val id = call.parameters["id"]?.toIntOrNull() // Safe conversion to Int
 if (id == null) {
 call.respond(HttpStatusCode.BadRequest, "Product ID must be an integer")
 return@get
 }

 val product = products.find { it.id == id }
 if (product != null) {
 call.respond(product)
 } else {
 call.respond(HttpStatusCode.NotFound, "Product not found")
 }
 }

 post("/products") {
 try {
 val product = call.receive<Product>()

```

```

// Simple validation (add more
robust validation as needed)
 if (products.any { it.id ==
product.id }) {

call.respond(HttpStatusCode.Conflict, "Product with
this ID already exists")
 return@post
 }

 products.add(product)

call.respond(HttpStatusCode.Created, product) //
Respond with 201 Created and the new product
 } catch (e: Exception) {
 // Handle JSON parsing errors
or other exceptions

call.respond(HttpStatusCode.BadRequest, "Invalid
product data: ${e.message}")
 }
}
}.start(wait = true)
}
...

```

Once again, Gemini has hallucinated, however both responses became more verbose and exact. For example, at the end of both responses, they added the “.start(wait=true),” so both models understand what is required to help a junior programmer.

## Task 10

|                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Prompt:</b> <pre>def is_even(n):<br/>    return n % 2 == 0<br/>Summary: returns true if input number n is even, false otherwise<br/><br/>Write a brief summary of the following python code:<br/>def reverse_words ( sentence ) :<br/>    return ' '.join ( sentence . split () [::-1])<br/>Summary:</pre>             |                                                                                                                                                                                                                                                                                                    |
| <b>Strategy:</b> Few-Shot Prompting                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                    |
| <b>Gemini 2.0 Flash</b>                                                                                                                                                                                                                                                                                                   | <b>Codestral 25.10</b>                                                                                                                                                                                                                                                                             |
| <pre>**Summary:**<br/><br/>The function `reverse_words` takes a sentence as input, splits it into individual words, reverses the order of the words, and then joins them back together into a new sentence, which is returned as the output. Effectively, it reverses the order of the words in the input sentence.</pre> | <pre>The provided Python code defines a function `reverse_words` that takes a `sentence` as input. The function splits the sentence into words, reverses the order of the words, and then joins them back together with spaces in between. The resulting reversed sentence is then returned.</pre> |

This Few-Shot prompting technique was very effective at keeping the responses short in this case as well as very accurate. It would work well as documentation.

|                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                                      |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Prompt:</b> <pre>You are a helpful coding assistant who summarises code into 1 or 2 lines.<br/><br/>Write a brief summary of the following python code:<br/>def reverse_words ( sentence ) :<br/>    return ' '.join ( sentence . split () [::-1])<br/>Summary:</pre> |                                                                                                                                                                                                                      |
| <b>Strategy:</b> Role-Play                                                                                                                                                                                                                                               |                                                                                                                                                                                                                      |
| <b>ChatGPT-4o</b>                                                                                                                                                                                                                                                        | <b>Gemini 2.0 Flash</b>                                                                                                                                                                                              |
| <pre>Reverses the order of words in a given sentence. It splits the sentence into words, reverses the order of the words, and joins them back into a string.</pre>                                                                                                       | <pre>The code defines a function `reverse_words` that takes a sentence, splits it into words, reverses the order of the words, and then joins them back into a single string, returning the reversed sentence.</pre> |

In this case, the Role-Play technique was just as, if not more, effective than the Few-Shot prompting at providing a brief summary. I think this prompt shows that adding context of specific length over a single word, like “brief,” is much more effective.

## Task 11

|                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Prompt:</b>                                                                                                                                                                                                                                                                                                                                          | Please write a prompt that would generate the following python code:<br># This function checks if a number is prime<br>def is_prime ( n ) :<br>if n <= 1:<br>return False<br>for i in range ( 2 , int( n **0.5) +1) :<br>if n % i == 0:<br>return False<br>return True                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Strategy:</b>                                                                                                                                                                                                                                                                                                                                        | Zero-Shot                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>ChatGPT-4o</b>                                                                                                                                                                                                                                                                                                                                       | <b>Codestral 25.10</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <pre>**Prompt:**<br/><br/>Write a Python function called is_prime that takes one argument n and returns True if n is a prime number, and False otherwise. The function should handle edge cases (numbers less than or equal to 1), and for efficiency, only check divisibility up to the square root of n. Include comments to explain the logic.</pre> | <pre>...<br/><br/>**Prompt:**<br/><br/>Write a Python function named `is_prime` that takes an integer `n` as input and returns a boolean value indicating whether `n` is a prime number. A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself. Here are the steps to implement the function:<br/><br/>1. If `n` is less than or equal to 1, return `False` because numbers less than or equal to 1 are not prime.<br/>2. Loop through all numbers from 2 up to the square root of `n` (inclusive) and check if `n` is divisible by any of these numbers.<br/>3. If `n` is divisible by any number in this range, return `False`.<br/>4. If no divisors are found, return `True`.<br/><br/>...</pre> |

The Zero-Shot prompt works to different effects here: ChatGPT provides a very human (NL) answer that would be more likely to come from a person, while Codestral provides a much more accurate and specific answer that is less likely to be written by a human, but more likely to provide the exact response.

| <b>Prompt:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Please write a prompt that would generate the following python code:<br><pre># This function checks if a number is prime def is_prime ( n ) :     if n &lt;= 1:         return False     for i in range ( 2 , int( n **0.5) +1) :         if n % i == 0:             return False     return True</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Strategy:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | <b>Self-Consistency</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| ChatGPT-4o                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Codestral 25.10                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <p><b>**Prompt:**</b></p> <p>Write a Python function called <code>is_prime</code> that checks whether a given integer <code>n</code> is a prime number. The function should return <code>False</code> if <code>n</code> is less than or equal to 1. For values greater than 1, check divisibility from 2 up to the square root of <code>n</code> (inclusive). If <code>n</code> is divisible by any of these numbers, return <code>False</code>; otherwise, return <code>True</code>. Include comments explaining that the function checks if a number is prime.</p> | <p>Sure! Here is a prompt that would generate the given Python code:</p> <p>---</p> <p>Write a Python function named <code>'is_prime'</code> that takes an integer <code>'n'</code> as an argument and returns <code>'True'</code> if <code>'n'</code> is a prime number, and <code>'False'</code> otherwise. A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself.</p> <p>Include the following logic in your function:</p> <ol style="list-style-type: none"> <li>1. If <code>'n'</code> is less than or equal to 1, return <code>'False'</code> because prime numbers are greater than 1.</li> <li>2. Use a loop to check for factors of <code>'n'</code> from 2 up to the square root of <code>'n'</code>. If <code>'n'</code> is divisible by any of these numbers, return <code>'False'</code>.</li> <li>3. If no factors are found, return <code>'True'</code>.</li> </ol> <p>---</p> <p>This prompt should guide the generation of the provided Python code.</p> |

The Self–Consistency prompt has a similar distinction between the two models, where ChatGPT is more human, while Codestral is more accurate. I do not think there is a significant difference between the use of either of these strategies, especially for this task.

## Task 12

### Prompt:

```
Please find the bug in the following python code snippet:
def factorial (n) :
 result = 1
 for i in range (1 , n) :
 result *= i
 return result

Explain your thought process step-by-step.
```

### Strategy:

### Chain of Thought

#### Gemini 2.0 Flash

```
...
1. Understanding the Goal
...
def factorial(n):
 result = 1
 for i in range(1, n + 1): # Include n in the
range
 result *= i
 return result
...
```

#### Codestral 25.10

```
...
1. **Understanding the Problem**:
...
def factorial(n):
 result = 1
 for i in range(1, n + 1):
 result *= i
 return result
...
```

The Chain of Thought strategy once again caused the unique line, “Understanding the ...,” which we saw in the Kotlin task. The actual code differences on such a minor bug fix are minimal, especially when compared to the next strategy on the same task.



| <b>Prompt:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Please find the bug in the following python code snippet:<br><pre>def factorial ( n ) :     result = 1     for i in range (1 , n ) :         result *= i     return result</pre><br>[Response 2]<br><br>Could you please rewrite the method such that the bug is fixed                                                                                                                        |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Strategy:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Prompt Chaining</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Gemini 2.0 Flash</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | <b>Codestral 25.10</b>                                                                                                                                                                                                                                                                                                                                                                        |
| <pre>... Could you please rewrite the method such that the bug is fixed  def factorial(n):     """     Calculates the factorial of a non-negative     integer.      Args:         n: A non-negative integer.      Returns:         The factorial of n. Returns 1 if n is 0.      Raises:         TypeError: if n is not an integer.         ValueError: if n is a negative integer.     """     if not isinstance(n, int):         raise TypeError("Input must be an integer.")     if n &lt; 0:         raise ValueError("Input must be a non-negative integer.")      result = 1     for i in range(1, n + 1):         result *= i     return result ...</pre> | <pre>... Could you please rewrite the method such that the bug is fixed  Certainly! Here is the corrected version of the `factorial` method with the bug fixed:  ```python def factorial(n):     result = 1     for i in range(1, n + 1):         result *= i     return result ```  This version ensures that the loop includes `n`, thereby correctly computing the factorial of `n`.</pre> |

Both models had a different response to the prompt chaining. As both had gotten off task and solved the bug in the initial response, Codestral just repeats itself while Gemini is much more expansive on the same code sample.

## Task 13

### Prompt:

Please implement node deletion by value in the following C code:

```
struct Node {
 int data ;
 struct Node * next ;
};
void deleteNode (struct Node ** head , int key) {
 // TODO : Implement node deletion
}
```

### Strategy:

### Zero-Shot

#### Codestral 25.10

```
...
#include <stdio.h>
#include <stdlib.h>

struct Node {
 int data;
 struct Node *next;
};

// Function to delete a node with a given key
void deleteNode(struct Node **head, int key) {
 // Store head node
 struct Node *temp = *head, *prev = NULL;

 // If head node itself holds the key to be deleted
 if (temp != NULL && temp->data == key) {
 *head = temp->next; // Changed head
 free(temp); // Free old head
 return;
 }

 // Search for the key to be deleted, keep track of
the // previous node as it is needed to change 'next'
 while (temp != NULL && temp->data != key) {
 prev = temp;
 temp = temp->next;
 }

 // If key was not present in the list
 if (temp == NULL) return;

 // Unlink the node from linked list
 prev->next = temp->next;

 free(temp); // Free memory
}

// Function to print the linked list
void printList(struct Node *node) {
 while (node != NULL) {
 printf(" %d ", node->data);
 node = node->next;
 }
 printf("\n");
}
...
```

#### Gemini 2.0 Flash

```
#include <stdio.h>
#include <stdlib.h>

struct Node {
 int data;
 struct Node *next;
};

void deleteNode(struct Node **head, int key) {
 // Store head node
 struct Node *temp = *head, *prev;

 // If head node itself holds the key to be
deleted
 if (temp != NULL && temp->data == key) {
 *head = temp->next; // Changed head
 free(temp); // Free old head
 return;
 }

 // Search for the key to be deleted, keep track
of the // previous node as we need to change
 'prev->next'
 while (temp != NULL && temp->data != key) {
 prev = temp;
 temp = temp->next;
 }

 // If key was not present in linked list
 if (temp == NULL)
 return;

 // Unlink the node from linked list
 prev->next = temp->next;

 // Free memory
 free(temp);
}

// A utility function to create a new node
struct Node *newNode(int data) {
 struct Node *temp = (struct Node
*)malloc(sizeof(struct Node));
 temp->data = data;
 temp->next = NULL;
 return temp;
}

// A utility function to print linked list
void printList(struct Node *head) {
 while (head != NULL) {
 printf("%d ", head->data);
 head = head->next;
 }
 printf("\n");
}
```

With Zero-Shot prompting, as has become apparent with the other prompts used, both Models stray from the original prompt and add extra code unelicited. Even with the more open ended problem, both Models gave the exact same result for the intended part of the problem.

### Prompt:

You are a helpful coding assistant who implements OOP code:

Please implement node deletion by value in the following C code:

```
struct Node {
 int data ;
 struct Node * next ;
};
void deleteNode (struct Node ** head , int key) {
 // TODO : Implement node deletion
}
```

### Strategy:

### Role-Play

#### Codestral 25.10

Here's an implementation of the `deleteNode` function in C to delete a node by its value. This function assumes that the list can be empty and that the node to be deleted may be the head of the list.

```
```c
#include <stdio.h>
#include <stdlib.h>

// Definition of the Node structure
struct Node {
    int data;
    struct Node *next;
};

// Function to delete a node with a given key
void deleteNode(struct Node **head, int key) {
    // Store the head node
    struct Node *temp = *head, *prev = NULL;

    // If head node itself holds the key to be deleted
    if (temp != NULL && temp->data == key) {
        *head = temp->next; // Changed head
        free(temp);        // Free old head
        return;
    }

    // Search for the key to be deleted, keep track of
    the previous node as it is needed to change 'next'
    while (temp != NULL && temp->data != key) {
        prev = temp;
        temp = temp->next;
    }

    // If key was not present in the linked list
    if (temp == NULL) return;

    // Unlink the node from the linked list
    prev->next = temp->next;

    free(temp); // Free the memory
}

// Function to print the linked list
void printList(struct Node *node) {
    while (node != NULL) {
        printf("%d ", node->data);
        node = node->next;
    }
    printf("\n");
}
```

Gemini 2.0 Flash

```
```c
#include <stdio.h>
#include <stdlib.h>

struct Node {
 int data ;
 struct Node * next ;
};

void deleteNode (struct Node ** head , int key) {
 // Store head node
 struct Node* temp = *head, *prev;

 // If head node itself holds the key to be
 deleted
 if (temp != NULL && temp->data == key) {
 *head = temp->next; // Changed head
 free(temp); // free old head
 return;
 }

 // Search for the key to be deleted, keep track
 of the
 // previous node as we need to change
 'prev->next'
 while (temp != NULL && temp->data != key) {
 prev = temp;
 temp = temp->next;
 }

 // If key was not present in linked list
 if (temp == NULL) return;

 // Unlink the node from linked list
 prev->next = temp->next;

 // Free memory
 free(temp);
}

// A utility function to create a new node
struct Node* newNode(int data)
{
 struct Node* node = (struct Node*)
 malloc(sizeof(struct Node));
 node->data = data;
 node->next = NULL;
 return node;
}
```

```

}
// Function to push a new node at the beginning of the list
void push(struct Node **head, int new_data) {
 struct Node *new_node = (struct Node *)malloc(sizeof(struct Node));
 new_node->data = new_data;
 new_node->next = (*head);
 (*head) = new_node;
}
...

// Function to print linked list
void printList(struct Node* head)
{
 while (head != NULL) {
 printf(" %d ", head->data);
 head = head->next;
 }
 ...
}

```

The Role-Play prompting strategy did not do much to reduce straying from the original task however Gemini did not add the newNode() function. The Role-Play strategy did decrease the length of the overall response, focusing only on the implementation over the explanation.

## Task 14

### Prompt:

```
def factorial(n):
 # TODO : Base cases and recursive call
 pass
COMPLETE CODE: return 1 if n <= 1 else n * factorial(n - 1)

Please complete the following python code:
def fibonacci (n) :
 # TODO : Base cases and recursive call
 pass
COMPLETE CODE:
```

### Strategy:

### Few-Shot Prompt

#### ChatGPT-4o

Certainly! Here is the completed code for the Fibonacci sequence using recursion:

```
```python
def fibonacci(n):
    # Base cases
    if n == 0:
        return 0
    elif n == 1:
        return 1
    # Recursive call
    else:
        return fibonacci(n - 1) + fibonacci(n - 2)
...

**COMPLETE CODE:**
```python
def fibonacci(n):
 return n if n <= 1 else fibonacci(n-1) +
 fibonacci(n-2)
...

```

#### Llama-4-Maverick

**\*\*Fibonacci Function in Python\*\***  
=====

Here's a Python function that calculates the nth Fibonacci number using recursion.

```
```python
def fibonacci(n):
    """
    Calculate the nth Fibonacci number.

    Args:
        n (int): The position of the Fibonacci number
        to calculate.

    Returns:
        int: The nth Fibonacci number.

    Raises:
        ValueError: If n is a negative integer.
    """

    # Check if n is a non-negative integer
    if not isinstance(n, int) or n < 0:
        raise ValueError("n must be a non-negative
        integer")

    # TODO: Base cases and recursive call
    # Base cases
    if n == 0:
        return 0
    elif n == 1:
        return 1
    # Recursive call
    else:
        return fibonacci(n-1) + fibonacci(n-2)
...

```

With this Few-Shot prompt strategy, it has a much larger effect on ChatGPT over Llama. ChatGPT even copied the one line code response that was given in the shot. Llama only expands on the code, and describes it in greater detail, rather than following the shot strategy.

Prompt:

Please complete the following python code:
 def fibonacci (n) :
 # TODO : Base cases and recursive call
 pass
 Explain your thought process step-by-step.

Strategy:**Chain of Thought****ChatGPT-4o**

```
## Step 1: Understand the Fibonacci Sequence

...

def fibonacci(n):
    # Base cases
    if n == 0:
        return 0
    elif n == 1:
        return 1
    # Recursive call
    else:
        return fibonacci(n - 1) + fibonacci(n - 2)
...
```

Llama-4-Maverick

```
...

1. **Understanding the Fibonacci Sequence**

...

## Optimizing the Fibonacci Function

The recursive implementation provided is straightforward but not efficient for large values of `n` due to the repeated computation of the same Fibonacci numbers. A more efficient solution would involve using dynamic programming techniques like memoization to store and reuse previously computed values.

Here's an example of how you can optimize it using memoization:

```python
def fibonacci(n, memo={}):
 """
 Calculate the nth Fibonacci number with memoization.

 Args:
 n (int): The position of the Fibonacci number to be calculated.
 memo (dict): A dictionary to store previously computed Fibonacci numbers.

 Returns:
 int: The nth Fibonacci number.

 Raises:
 ValueError: If n is a negative integer.
 """
 if n < 0:
 raise ValueError("n must be a non-negative integer.")

 # Base cases
 if n == 0:
 return 0
 elif n == 1:
 return 1

 # Check if the Fibonacci number is already computed
 elif n in memo:
 return memo[n]

 # Compute and store the Fibonacci number
 else:
 result = fibonacci(n-1, memo) + fibonacci(n-2, memo)
 memo[n] = result
 return result
...
```

Using the Chain of Thought strategy, it can be observed that Llama behaves more creatively and efficiently than ChatGPT. Without the guidance, ChatGPT answers normally, however, with the Chain of Thought prompt causes Llama to provide a much more efficient solution using a

dictionary. Once again it can also be observed that the models use the “Understanding the ...” phrase.

## Task 15

### Prompt:

```
class Spaceship:
 def __init__(self):
 # TODO : Add name and fuel
 pass

COMPLETE CODE:
 self.name = name
 self.fuel = fuel

Please complete the following __init__ method for the python class
class Person :
 def __init__ (self) :
 # TODO : Add name , age , and optional email
 pass

COMPLETE CODE:
```

### Strategy:

### Few-Shot Prompt

#### Codestral 25.10

Certainly! Below is the completed `\_\_init\_\_` method for the `Person` class, which includes `name`, `age`, and an optional `email` parameter:

```
```python
class Person:
    def __init__(self, name, age, email=None):
        self.name = name
        self.age = age
        self.email = email
```
```

This method initializes a `Person` object with a `name`, `age`, and an optional `email`. If no `email` is provided, it defaults to `None`.

#### Gemini 2.0 Flash

```
```python
class Spaceship:
    def __init__(self, name, fuel):
        # TODO: Add name and fuel
        self.name = name
        self.fuel = fuel

class Person:
    def __init__(self, name, age, email=None):
        # TODO : Add name , age , and optional
        email
        self.name = name
        self.age = age
        self.email = email
```
```

The two responses here are very similar for the Few-Shot prompt. Codestral is more concise but Gemini included the shot class. Overall, both responses complete the task however Gemini treats the shot as another task to complete.



**Prompt:**

Please complete the following `__init__` method for the python class

```
class Person :
 def __init__ (self) :
 # TODO : Add name , age , and optional email
 pass
```

**Strategy:****Self-Consistency****Codestral 25.10**

Sure, to complete the `__init__` method for the `Person` class, you can define parameters for the name, age, and an optional email. Here is an example of how you can do it:

```
```python
class Person:
    def __init__(self, name, age, email=None):
        self.name = name
        self.age = age
        self.email = email
```

```
# Example usage:
person1 = Person("Alice", 30)
person2 = Person("Bob", 25, "bob@example.com")
```

```
print(person1.name) # Output: Alice
print(person1.age)  # Output: 30
print(person1.email) # Output: None
```

```
print(person2.name) # Output: Bob
print(person2.age)  # Output: 25
print(person2.email) # Output: bob@example.com
```
```

In this implementation:

- The `__init__` method takes three parameters: `name`, `age`, and an optional `email`.
- `name` and `age` are required parameters, meaning you must provide values for them when creating an instance of the `Person` class.
- `email` is an optional parameter, and its default value is `None`. You can provide a value for it if you have an email address to include.

**Gemini 2.0 Flash**

```
```python
class Person:
    def __init__(self, name, age, email=None):
        """
        Initializes a Person object.

        Args:
            name (str): The person's name.
            age (int): The person's age.
            email (str, optional): The person's
            email address. Defaults to None.
        """
        self.name = name
        self.age = age
        self.email = email
...
```
```

**\*\*Explanation:\*\***

1. **\*\*Parameters:\*\*** The `__init__` method now takes three parameters:

- `name`: The person's name (required).
- `age`: The person's age (required).
- `email`: The person's email address (optional). We set a default value of `None` so that the email is not a required parameter.

2. **\*\*Attributes:\*\*** Inside the `__init__` method, we assign the values passed as arguments to the object's attributes using `self`:

- `self.name = name`
- `self.age = age`
- `self.email = email`

**\*\*How to use it:\*\***

```
```python
# Create a person with a name, age, and email
person1 = Person("Alice", 30, "alice@example.com")

# Create a person with just a name and age (no email)
person2 = Person("Bob", 25)

print(person1.name) # Output: Alice
print(person1.age)  # Output: 30
print(person1.email) # Output: alice@example.com

print(person2.name) # Output: Bob
print(person2.age)  # Output: 25
print(person2.email) # Output: None
```
```

Compared to the Few-Shot prompt, the Self-Consistency strategy elicits even more similar responses. Very interestingly, the driver code for testing used by both Models is so similar it suggests that the two Models have derived their answer from the exact same part of their corpus.

## Task 16

### Prompt:

```
Please complete the following Java implementation of a binary search:
public int binarySearch (int [] arr , int target) {
 int left = 0 , right = arr . length - 1;
 while (left <= right) {
 int mid = (left + right) / 2;
 // TODO : Compare and adjust bounds
 }
 return -1;
}
```

### Strategy:

### Zero-Shot

#### ChatGPT-4o

```
...
public int binarySearch(int[] arr, int target) {
 int left = 0, right = arr.length - 1;
 while (left <= right) {
 int mid = left + (right - left) / 2; // Avoids
overflow
 if (arr[mid] == target) {
 return mid; // Element
found
 } else if (arr[mid] < target) {
 left = mid + 1; // Search right
half
 } else {
 right = mid - 1; // Search left
half
 }
 }
 return -1; // Not found
}
...
```

#### Codestral 25.10

```
...
public int binarySearch(int[] arr, int target) {
 int left = 0, right = arr.length - 1;
 while (left <= right) {
 int mid = left + (right - left) / 2; // To
avoid overflow
 // Check if the target is present at mid
 if (arr[mid] == target) {
 return mid;
 }
 // If target is greater, ignore the left
half
 if (arr[mid] < target) {
 left = mid + 1;
 }
 // If target is smaller, ignore the right
half
 else {
 right = mid - 1;
 }
 }
 // Target is not present in the array
 return -1;
}
...
```

The difference between the two responses is merely superficial. Codestral saves a word by removing the else on their second if statement. Both responses are correct. The Zero-Shot prompt is overall very effective for this simple task.

**Prompt:**

You are a master hacker who is forced to complete code for junior programmers

Please complete the following Java implementation of a binary search:

```
public int binarySearch (int [] arr , int target) {
 int left = 0 , right = arr . length - 1;
 while (left <= right) {
 int mid = (left + right) / 2;
 // TODO : Compare and adjust bounds
 }
 return -1;
}
```

**Strategy:****Role-Play****ChatGPT-4o**

```
...
public int binarySearch(int[] arr, int target) {
 int left = 0, right = arr.length - 1;
 while (left <= right) {
 int mid = left + (right - left) / 2; //
Prevent potential overflow
 if (arr[mid] == target) {
 return mid; // found target
 } else if (arr[mid] < target) {
 left = mid + 1; // search right half
 } else {
 right = mid - 1; // search left half
 }
 }
 return -1; // not found
}
...
```

**Codestral 25.10**

```
...
public int binarySearch(int[] arr, int target) {
 int left = 0, right = arr.length - 1;

 while (left <= right) {
 int mid = left + (right - left) / 2; // To
avoid potential overflow

 // Check if the target is present at mid
 if (arr[mid] == target) {
 return mid;
 }

 // If target is greater, ignore the left
half
 if (arr[mid] < target) {
 left = mid + 1;
 }
 // If target is smaller, ignore the right
half
 else {
 right = mid - 1;
 }
 }

 // Target is not present in the array
 return -1;
}
...
```

The somewhat silly role for this Role-Play strategy was intended to see whether or not a stranger role would affect the model responses. Neither model seemed to be affected, and even the minor difference in their response remained the same, suggesting that a more specific role might have less effect.

## Task 17

| <b>Prompt:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Please suggest a more correct name for the following C++ function:<br>// Supposed to return true if x is even<br>bool isOdd (int x ) {<br>return x % 2 == 0; // Logic contradicts function name<br>}                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Strategy:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Zero-Shot                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Gemini 2.0 Flash                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Llama-4-Maverick                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <p>The most straightforward and correct name would be:</p> <pre>*  `isEven`</pre> <p>This aligns perfectly with the function's logic:<br/>`return x % 2 == 0;` which evaluates to `true` when<br/>`x` is divisible by 2 (i.e., even).</p> <p>Here are a few other options, although `isEven` is the best:</p> <pre>*  `isEvenNumber` (More verbose, but emphasizes it's checking a number)<br/>*  `isDivisibleByTwo` (Very explicit about the divisibility condition)</pre> | <pre>### Suggested Function Name</pre> <p>The function name `isOdd` is misleading because it returns `true` when the input `x` is even, not odd. A more correct name for this function would be `isEven`.</p> <pre>```cpp<br/>// Returns true if x is even<br/>bool isEven(int x) {<br/>    return x % 2 == 0;<br/>}<br/>```</pre> <pre>### Alternative Solution</pre> <p>Alternatively, you could keep the original function name `isOdd` and correct the logic inside the function to return `true` when `x` is odd.</p> <pre>```cpp<br/>// Returns true if x is odd<br/>bool isOdd(int x) {<br/>    return x % 2 != 0;<br/>}<br/>```</pre> <pre>### Best Practice</pre> <p>It's essential to ensure that function names accurately reflect their behavior to avoid confusion and make the code more maintainable. In this case, either renaming the function to `isEven` or correcting its logic to match the name `isOdd` is a good practice.</p> |

The Zero-Shot strategy for this particular task is clearly effective, especially when given a simple method. That being said, Llama veers especially off task, trying to suggest manners in which to change the method to fit the name, which was not included as part of the prompt definition.

| <p><b>Prompt:</b></p> <pre>// Supposed to return a - b int add(int a, int b) {     return a - b; } Correct Name: subtract  Please suggest a more correct name for the following C++ function: // Supposed to return true if x is even bool isOdd (int x ) {     return x % 2 == 0; // Logic contradicts function name } Correct Name:</pre> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Strategy:                                                                                                                                                                                                                                                                                                                                   | Few-Shot Prompt                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Gemini 2.0 Flash                                                                                                                                                                                                                                                                                                                            | Llama-4-Maverick                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <pre>Correct Name: **isEven**</pre>                                                                                                                                                                                                                                                                                                         | <pre>The correct name for the given function should be `isEven`.  The function is currently named `isOdd`, but it returns `true` if `x` is even, not odd. The condition `x % 2 == 0` checks if a number is even. To fix this, either the function name or the return condition should be corrected.  Here's the corrected version with the function name changed:  ```cpp // Returns true if x is even bool isEven(int x) {     return x % 2 == 0; } ```  Alternatively, you could change the return condition to correctly implement `isOdd`:  ```cpp // Returns true if x is odd bool isOdd(int x) {     return x % 2 != 0; } ```</pre> |

The Few-Shot strategy here is clearly most effective for Gemini, as it follows the given structure and gives the short, but intended, answer. While Llama is affected by coming to their final answer earlier, the contents of their response is the same and still veers off course.

## Task 18

| <b>Prompt:</b> Please identify and fix the following JavaScript bug:<br><pre>function isEven ( n ) {<br/>    return n % 2; // Returns 1 or 0 , not true / false<br/>}</pre><br>Explain your thought process step-by-step.                                                                                   |                                                                                                                                                                       |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Strategy:</b> Chain of Thought                                                                                                                                                                                                                                                                           |                                                                                                                                                                       |
| Llama-4-Maverick                                                                                                                                                                                                                                                                                            | Codestral 25.10                                                                                                                                                       |
| <pre>### Step-by-Step Analysis<br/>1. **Understanding the function's purpose**:<br/>...<br/>function isEven(n) {<br/>    // Using the modulus operator to check if n is<br/>    even.<br/>    // We compare the result to 0 to get a boolean<br/>    value.<br/>    return n % 2 === 0;<br/>}<br/>...</pre> | <pre>### Step-by-Step Analysis<br/>...<br/>function isEven(n) {<br/>    return n % 2 === 0; // Returns true if n is<br/>    even, false otherwise<br/>}<br/>...</pre> |

Besides how both models obtained the correct solution using Chain of Thought, I find it interesting to highlight the use of the phrase “Step-by-Step” for both models which is used by Llama for the next task, but while the “Understanding the ...” phrase is only used in one of the responses, it is unique to the Chain of Thought task and (although not shown) is not used in the following response.

**Prompt:**

```
Please identify the following bug:
function isEven (n) {
 return n % 2; // Returns 1 or 0 , not true / false
}
```

[Response 1]

Could you recommend a fix for the above bug?

**Strategy:****Prompt Chaining****Llama-4-Maverick****Codestral 25.10**

...

Could you recommend a fix for the above bug?

The bug can be fixed by modifying the return statement to return a boolean value. Here's the corrected code:

```
function isEven(n) {
 return n % 2 === 0; // Returns true if n is even,
 false otherwise
}
```

Alternatively, you can also use:

```
function isEven(n) {
 return !(n % 2); // Returns true if n is even,
 false otherwise
}
```

...

Could you recommend a fix for the above bug?

Certainly! To fix the bug in the `isEven` function so that it returns `true` or `false` instead of `1` or `0`, you need to compare the result of `n % 2` to `0`. Here is the corrected version of the function:

```
```javascript
function isEven(n) {
    return n % 2 === 0;
}
```
```

### Explanation:

- `n % 2` computes the remainder when `n` is divided by `2`.
- `n % 2 === 0` checks if the remainder is `0`, which means `n` is even.
- If `n` is even, the expression `n % 2 === 0` evaluates to `true`.
- If `n` is odd, the expression `n % 2 === 0` evaluates to `false`.

### Example Usage:

```
console.log(isEven(4)); // Outputs: true
console.log(isEven(7)); // Outputs: false
```

Prompt Chaining, especially in this particular case, caused Llama to think of other solutions, not just one. This suggests an additional level of creativity elicited by the prompt. On the other hand, Codestral mostly repeated its previous response, which included the answer.

## Task 19

| <b>Prompt:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Please decompose the following C++ code segment into a high-level summary of what is happening:<br><pre>// Function that validates an input , calculates square , and returns result int process (int x ) {     if ( x &lt; 0) return -1;     return x * x ; }</pre>                                                                                                                                                                                                                           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Strategy:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Zero-Shot                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Llama-4-Maverick                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Gemini 2.0 Flash                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <pre>**Code Summary**  The given C++ code defines a function `process` that takes an integer `x` as input and performs the following operations:  1. <b>Input Validation</b>: Checks if the input `x` is negative. 2. <b>Error Handling</b>: If `x` is negative, returns `-1` to indicate an invalid input. 3. <b>Square Calculation</b>: If `x` is non-negative, calculates the square of `x`. 4. <b>Result Return</b>: Returns the result of the square calculation.  **In Summary**  The `process` function validates the input, calculates the square of a non-negative integer, and returns the result. If the input is negative, it returns `-1` to indicate an error.</pre> | <pre>This C++ code segment defines a function named `process` that takes an integer `x` as input. The function first checks if the input integer `x` is negative. If it is, the function immediately returns -1, indicating an invalid input. Otherwise, if `x` is non-negative, the function calculates the square of `x` (x * x) and returns the resulting integer value. In essence, the function either returns the square of a non-negative integer or -1 if the input is negative.</pre> |

This Zero-Shot prompt gives a helpful baseline with which to check how to model performs in the following prompt. Llama was un-human and included a section on *Error Handling* even though there were no exceptions or errors to be raised. It would have been more accurately called *Input Sanitization*. Gemini responded very human-like, making a document level code comment.



|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Prompt:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | You are a helpful coding assistant which summarises code                                                                                                                                                                                                                                                                                                                                                            |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Please decompose the following C++ code segment into a high-level summary of what is happening:<br>// Function that validates an input , calculates square , and returns result<br>int process (int x ) {<br>if ( x < 0) return -1;<br>return x * x ;<br>}                                                                                                                                                          |
| <b>Strategy:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | <b>Role-Play</b>                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Llama-4-Maverick</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | <b>Gemini 2.0 Flash</b>                                                                                                                                                                                                                                                                                                                                                                                             |
| <p><b>**Code Summary**</b></p> <p>The given C++ code defines a function `process` that takes an integer input `x`, validates it, and returns the square of `x` if valid.</p> <p><b>**Step-by-Step Breakdown**</b></p> <ol style="list-style-type: none"><li><b>**Input Validation**:</b> The function checks if the input `x` is less than 0.</li><li><b>**Error Handling**:</b> If `x` is negative, the function returns -1, indicating an invalid input.</li><li><b>**Square Calculation**:</b> If `x` is non-negative, the function calculates the square of `x` by multiplying it with itself (`x * x`).</li><li><b>**Result Return**:</b> The result of the square calculation is returned by the function.</li></ol> <p><b>**In Summary**</b>, the `process` function validates a non-negative integer input and returns its square, or -1 if the input is invalid (negative).</p> | <p>The C++ code defines a function named `process` that takes an integer `x` as input. It checks if `x` is negative. If `x` is negative, the function immediately returns -1. Otherwise, it calculates the square of `x` (<math>x * x</math>) and returns the result. In essence, the function returns the square of a non-negative input, and returns -1 if the input is negative, signaling an invalid input.</p> |

Compared to the Zero-Shot, this Role-Play strategy has no unique additions to either Model’s responses. I did try out a more general Role (coding assistant which summarises) which could have played a role in its decreased effectiveness especially when compared to some other Role-Play strategies.

## Task 20

|                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Prompt:</b>                                                                                                                                                                                                                                                                                                                      | Complete the python function based on the shown intent:<br><pre>def calculate_average ( scores ) :<br/>    total = 0<br/>    # TODO : Complete to return average<br/>    pass</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Strategy:</b>                                                                                                                                                                                                                                                                                                                    | <b>Zero-Shot</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Gemini 2.0 Flash</b>                                                                                                                                                                                                                                                                                                             | <b>Codestral 25.10</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <pre>```python<br/>def calculate_average ( scores ) :<br/>    total = 0<br/>    # TODO : Complete to return average<br/>    if not scores:<br/>        return 0 # Handle empty list case to avoid<br/>ZeroDivisionError<br/><br/>    total = sum(scores)<br/>    average = total / len(scores)<br/>    return average<br/>```</pre> | <pre>Certainly! To complete the function<br/>`calculate_average`, you need to iterate through<br/>the list of scores, sum them up, and then compute<br/>the average. Here's the completed function:<br/><br/>```python<br/>def calculate_average(scores):<br/>    total = 0<br/>    # Sum all the scores<br/>    for score in scores:<br/>        total += score<br/>    # Calculate the average<br/>    average = total / len(scores)<br/>    return average<br/>```<br/><br/>Here's a brief explanation of what each part does:<br/>1. <b>Initialization</b>: You start with `total` set<br/>to 0.<br/>2. <b>Summing Scores</b>: The `for` loop iterates<br/>through each `score` in the `scores` list and adds<br/>it to `total`.<br/>3. <b>Calculating Average</b>: After summing all the<br/>scores, you divide `total` by the number of scores<br/>(which is `len(scores)`) to get the average.<br/>4. <b>Returning the Average</b>: Finally, the function<br/>returns the calculated average.<br/><br/>Here is an example of how you might use this<br/>function:<br/><br/>```python<br/>scores = [85, 90, 78, 92, 88]<br/>average = calculate_average(scores)<br/>print("The average score is:", average)<br/>```<br/><br/>This will output:<br/>```<br/>The average score is: 86.6<br/>```</pre> |

The response to the Zero-Shot strategy by Gemini is more concise and effective code, a response more expected at this point to come from Few-Shot prompting, while that of Codestral is further explained. Gemini also answers the method in a more experienced manner, utilising built-in python functions and includes input sanitization, while Codestral codes in a more junior manner, that works for most inputs.

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Prompt:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Complete the python function based on the shown intent:<br><pre>def calculate_average ( scores ) :<br/>    total = 0<br/>    # TODO : Complete to return average<br/>    pass</pre><br>Explain your thought process step-by-step.                                                                                                                                                                                                                                                                                                                          |
| <b>Strategy:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                  | <b>Chain of Thought</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Gemini 2.0 Flash</b>                                                                                                                                                                                                                                                                                                                                                                                                                                           | <b>Codestral 25.10</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <pre>Okay, let's break down how to complete this Python function.<br/><br/>**Understanding the Goal**<br/><br/>...<br/><br/>**Python Code**<br/><br/>def calculate_average ( scores ) :<br/>    total = 0<br/>    # TODO : Complete to return average<br/>    if not scores: # Check if the list is empty<br/>        return 0<br/>    for score in scores:<br/>        total += score<br/>    average = total / len(scores)<br/>    return average<br/>...</pre> | <pre>...<br/><br/>def calculate_average(scores):<br/>    total = 0<br/>    # Sum all the scores<br/>    for score in scores:<br/>        total += score<br/><br/>    # Calculate the average<br/>    average = total / len(scores)<br/><br/>    # Return the average<br/>    return average<br/><br/>...<br/><br/>3. **Compute the Average**:<br/>    ```python<br/>    average = total / len(scores)<br/>    ```<br/><br/>    Here, `total` is divided by the number of scores<br/>    (`len(scores)`), which gives the average score.<br/><br/>...</pre> |

With the Chain of Thought prompting, Codestral still fails to sanitize inputs for an empty list. I have left in its explanation of the average calculation line, and it shows no clear thought for looking for errors. I have also left in the header of “Understanding the ...” which appears in all the other models’ Chain of Thought responses except for Codestral.

## Task 21

### Prompt:

```
Please analyze the following python code and identify any logic issues:
utils.py - Script to parse and summarize numeric CSV files
import csv
def read_csv (filepath) :
 with open (filepath , 'r') as f :
 return [row for row in csv . reader (f)]
def summarize_column (data , index) :
 values = [float (row [index]) for row in data [1:]] # skip header
 total = sum(values)
 avg = total / len (values)
 return total , avg
def main () :
 filepath = 'data .csv '
 data = read_csv (filepath)
 total , avg = summarize_column (data , 1)
 print (" Total :", total)
 print (" Average :", avg)
if __name__ == '__main__':
 main ()

[Response 1]

Could you generate the refactored code for better readability, correctness, and security?
```

### Strategy:

### Prompt Chaining

#### Codestral 25.10

The provided Python code has several issues and potential areas for improvement. Let's go through them step-by-step:

- \*\*Indentation Error in `read\_csv` Function\*\*:**
  - The `return` statement inside the `with` block is not properly indented. This will result in a syntax error.
- \*\*File Path String Issue\*\*:**
  - The `filepath` string has an extra space (`'data .csv '`). It should be `'data.csv'`.
- \*\*Error Handling\*\*:**
  - There is no error handling for file operations. If the file does not exist or cannot be opened, the code will raise an exception.
- \*\*Type Error in `summarize\_column` Function\*\*:**
  - If any of the rows in the CSV file cannot be converted to a float, this will raise a `ValueError`. This should be handled.
- \*\*Column Indexing\*\*:**
  - The code assumes that the CSV file has a header row and that the index for summarization is always 1. This might not be true for all CSV files.
- \*\*Code Formatting\*\*:**
  - The code formatting is inconsistent. For example, spaces around parentheses and after commas are missing.

...

Could you generate the refactored code for better readability, correctness, and security?

Sure! Here is the refactored code with improved readability, correctness, and security:

```
utils.py - Script to parse and summarize numeric CSV files
```

#### Llama-4-Maverick

...

#### #### Code Structure and Readability

- \* The code is generally well-structured, with separate functions for reading the CSV file and summarizing a column.
- \* However, there are some formatting issues, such as inconsistent indentation and missing whitespace around operators.

#### #### Potential Logic Issues

- \*\*Assumes a header row\*\*:** The `summarize\_column` function skips the first row of the data, assuming it's a header. However, this assumption is not validated. If the CSV file doesn't have a header row, the first data row will be skipped.
- \*\*No error handling for file operations\*\*:** The `read\_csv` function does not handle potential errors when opening or reading the file, such as file not found or permission denied errors.
- \*\*No validation for CSV data\*\*:** The `summarize\_column` function assumes that the specified column contains numeric data. If the column contains non-numeric data, a `ValueError` will be raised when trying to convert it to a float.
- \*\*Division by zero\*\*:** If the specified column is empty (i.e., all rows have a missing or non-numeric value), `len(values)` will be zero, and a `ZeroDivisionError` will occur when calculating the average.
- \*\*Index-based column selection\*\*:** The `summarize\_column` function uses an index to select the column to summarize. This can be error-prone if the CSV file structure changes. Consider using a column name instead.
- \*\*Hardcoded file path\*\*:** The `main` function uses a hardcoded file path. Consider making the file path a command-line argument or a configurable parameter.
- \*\*Lack of input validation\*\*:** The

```

import csv

def read_csv(filepath):
 """
 Reads a CSV file and returns its contents as a
 list of rows.
 """
 try:
 with open(filepath, 'r', newline='') as file:
 reader = csv.reader(file)
 return list(reader)
 except FileNotFoundError:
 print(f"Error: The file {filepath} does not
 exist.")
 return []
 except Exception as e:
 print(f"An error occurred while reading the
 file: {e}")
 return []

def summarize_column(data, index):
 """
 Summarizes a column in the CSV data by calculating
 the total and average of the numeric values.
 """
 if not data:
 print("No data to summarize.")
 return None, None

 try:
 values = [float(row[index]) for row in
 data[1:]] # skip header
 total = sum(values)
 average = total / len(values)
 return total, average
 except (ValueError, IndexError) as e:
 print(f"Error: {e}")
 return None, None

def main():
 """
 Main function to execute the CSV parsing and
 summarization.
 """
 filepath = 'data.csv'
 data = read_csv(filepath)
 if data:
 total, avg = summarize_column(data, 1)
 if total is not None and avg is not None:
 print("Total:", total)
 print("Average:", avg)

if __name__ == '__main__':
 main()
...

```

`summarize\_column` function does not validate its inputs. For example, it assumes that `index` is a valid column index.

...

Could you generate the refactored code for better readability, correctness, and security?

### Refactored Code

Here's a refactored version of the provided Python script with improved readability, correctness, and security:

```

import csv
import argparse
import logging

Configure logging
logging.basicConfig(level=logging.INFO,
 format='%(asctime)s - %(levelname)s - %(message)s')

def read_csv(filepath):
 """
 Reads a CSV file and returns its contents as a
 list of lists.

 Args:
 filepath (str): Path to the CSV file.

 Returns:
 list: A list of lists, where each inner
 list represents a row in the CSV file.
 """
 try:
 with open(filepath, 'r') as file:
 csv_reader = csv.reader(file)
 data = list(csv_reader)
 return data
 except FileNotFoundError:
 logging.error(f"File '{filepath}' not
 found.")
 return []
 except Exception as e:
 logging.error(f"An error occurred: {e}")
 return []

def summarize_column(data, column_name):
 """
 Summarizes a column in the provided CSV data.

 Args:
 data (list): A list of lists, where each
 inner list represents a row in the CSV file.
 column_name (str): The name of the column
 to summarize.

 Returns:
 tuple: A tuple containing the total and
 average of the specified column.
 """
 if not data or len(data) < 2:
 logging.error("Insufficient data.")
 return None, None

 header = data[0]
 try:
 index = header.index(column_name)
 except ValueError:
 logging.error(f"Column '{column_name}' not
 found.")
 return None, None

 values = []
 for row in data[1:]:
 try:
 value = float(row[index])
 values.append(value)
 except (IndexError, ValueError):

```

```

 logging.warning(f"Non-numeric or
missing value in column '{column_name}'.")

 if not values:
 logging.error("No numeric values in
column.")
 return None, None

 total = sum(values)
 avg = total / len(values)
 return total, avg

def main():
 parser =
 argparse.ArgumentParser(description='Summarize a
CSV column.')
 parser.add_argument('filepath', help='Path to
the CSV file.')
 parser.add_argument('column_name', help='Name
of the column to summarize.')
 args = parser.parse_args()

 data = read_csv(args.filepath)
 total, avg = summarize_column(data,
args.column_name)
 if total is not None and avg is not None:
 logging.info(f"Total: {total}")
 logging.info(f"Average: {avg}")

if __name__ == '__main__':
 main()

...

```

Even with a longer and more challenging response, both Models still went off-course and suggested a code fix after the first prompt. I have kept only the fix after the second prompt to look for any improvement. Their suggested potential errors both included some of the same basic responses (lack of error handling, general syntax issues, column indexing) but Llama showed a greater awareness of potential logic issues and bad practice (hard coded file paths, assumes header row). Their actual code is similar however Llama finds it necessary to include additional python modules not originally in use, actually adding the logging module in their second response as it wasn't in the first.

**Prompt:**

Please analyze the following python code and identify any logic issues then generate the refactored code to improve readability, correctness, and security:

```
utils.py - Script to parse and summarize numeric CSV files
import csv
def read_csv (filepath) :
 with open (filepath , 'r') as f :
 return [row for row in csv . reader (f)]
def summarize_column (data , index) :
 values = [float (row [index]) for row in data [1:]] # skip header
 total = sum (values)
 avg = total / len (values)
 return total , avg
def main () :
 filepath = 'data .csv '
 data = read_csv (filepath)
 total , avg = summarize_column (data , 1)
 print (" Total :", total)
 print (" Average :", avg)
if __name__ == '__main__' :
 main ()
```

Explain your thought process step-by-step.

**Strategy:****Chain of Thought****Codestral 25.10**

Let's analyze the provided Python code step-by-step to identify any logic issues and improve its readability, correctness, and security.

...

Let's refactor the code to address these issues:

```
utils.py - Script to parse and summarize numeric CSV files
import csv
```

```
def read_csv(filepath):
 try:
 with open(filepath, 'r') as f:
 return [row for row in csv.reader(f)]
 except FileNotFoundError:
 print(f"Error: The file {filepath} does not exist.")
 return []
 except Exception as e:
 print(f"Error reading the file {filepath}: {e}")
 return []
```

```
def summarize_column(data, column_name):
 if not data:
 print("Error: No data to summarize.")
 return None, None
```

```
 header = data[0]
 if column_name not in header:
 print(f"Error: Column '{column_name}' does not exist in the data.")
 return None, None
```

```
 index = header.index(column_name)
 values = []
 for row in data[1:]:
 try:
 values.append(float(row[index]))
 except (ValueError, IndexError):
 print(f"Warning: Non-numeric value found in row {data.index(row)}.")
 continue
```

```
 if not values:
 print(f"Error: No valid numeric values found in column '{column_name}'.")
 return None, None
```

**Llama-4-Maverick**

**\*\*Step 1: Initial Review\*\***

...

```
import csv
import argparse
```

```
def read_csv(filepath):
 """
 Reads a CSV file into a list of rows.

 Args:
 filepath (str): Path to the CSV file.

 Returns:
 list: List of rows in the CSV file.
```

```
 Raises:
 FileNotFoundError: If the file does not exist.
 csv.Error: If the file is not a valid CSV file.
```

```
 """
 try:
 with open(filepath, 'r') as f:
 return list(csv.reader(f))
 except FileNotFoundError:
 print(f"Error: File '{filepath}' not found.")
 return []
 except csv.Error as e:
 print(f"Error: Failed to parse CSV file '{filepath}': {e}")
 return []
```

```
def summarize_column(data, index):
 """
 Calculates the total and average of a column in the given data.
```

```
 Args:
 data (list): List of rows in the CSV file.
 index (int): Index of the column to summarize.
```

```
 Returns:
 tuple: Total and average of the column.
```

```
 Raises:
 ValueError: If the column index is out of bounds or if the data is not numeric.
```

```

total = sum(values)
avg = total / len(values)
return total, avg

def main():
 filepath = 'data.csv'
 data = read_csv(filepath)
 if not data:
 return

 column_name = 'your_column_name' # Change this to
 the column you want to summarize
 total, avg = summarize_column(data, column_name)
 if total is not None and avg is not None:
 print(f"Total: {total}")
 print(f"Average: {avg}")

if __name__ == '__main__':
 main()

Explanation of Changes:
...
"""
if index < 0 or index >= len(data[0]):
 raise ValueError("Column index is out of
 bounds")

 try:
 values = [float(row[index]) for row in
 data[1:]] # skip header
 total = sum(values)
 avg = total / len(values) if values else 0
 return total, avg
 except ValueError as e:
 raise ValueError("Non-numeric data in
 column") from e

def main():
 parser =
 argparse.ArgumentParser(description="Summarize a
 numeric CSV column")
 parser.add_argument("filepath", help="Path to
 the CSV file")
 parser.add_argument("-c", "--column", type=int,
 default=1, help="Column index to summarize
 (0-based)")
 args = parser.parse_args()

 data = read_csv(args.filepath)
 if not data:
 return

 try:
 total, avg = summarize_column(data,
 args.column)
 print(f"Total: {total}")
 print(f"Average: {avg}")
 except ValueError as e:
 print(f"Error: {e}")

if __name__ == '__main__':
 main()

Changes and Improvements
...

```

Looking at a more complicated task using the Chain of Thought strategy, it did not actually differ too far from the first half of the prompt chaining strategy. I have included the headers for both models where they would overview changes to be made and then changes that have been made, but this is done in a very similar manner in the previous prompting strategy for this task, which was not true in simpler tasks when comparing Chain of Thought to other strategies. For the actual answer, once again Llama utilises extra dependencies, however, makes a clearly more detailed response, but no more detailed than the previous strategy.



## Task 22

### Prompt:

Could you please complete the following python script so that it cleans each line, removes punctuation, and counts word frequency:

```
file_processor.py - Incomplete script for processing text files
import string
def load_file (filepath) :
 with open (filepath , 'r') as f :
 return f . readlines ()
def clean_line (line) :
 # TODO : Remove punctuation and make lowercase
 pass
def count_words (lines) :
 word_counts = {}
 for line in lines :
 clean = clean_line (line)
 for word in clean . split () :
 word_counts [word] = word_counts . get (word , 0) + 1
 return word_counts
def main () :
 filepath = 'input.txt '
 lines = load_file (filepath)
 counts = count_words (lines)
 for word , count in sorted (counts . items ()) :
 print (f"{ word } : { count }")
if __name__ == '__main__' :
 main ()
```

### Strategy:

### Zero-Shot

#### Gemini 2.0 Flash

#### Codestral 25.10

```
```python
# file_processor.py - Incomplete script for
processing text files
import string

def load_file ( filepath ) :
    with open ( filepath , 'r') as f :
        return f . readlines ()

def clean_line ( line ) :
    # Remove punctuation and make lowercase
    line = line.lower()
    line = line.translate(str.maketrans('', '',
string.punctuation))
    return line

def count_words ( lines ) :
    word_counts = {}
    for line in lines :
        clean = clean_line ( line )
        for word in clean . split () :
            word_counts [ word ] = word_counts . get (
word , 0) + 1
    return word_counts

def main () :
    filepath = 'input.txt' # Corrected filepath
    lines = load_file ( filepath )
    counts = count_words ( lines )
    for word , count in sorted ( counts . items () ) :
        print ( f"{ word } : { count }")

if __name__ == '__main__':
    main ()
...`
```

Key improvements and explanations:

```
* **`clean_line(line)` function:** This is the core
of the solution.
* **`line.lower()`:** Converts the input `line` to
lowercase. This ensures that words like "The" and
"the" are counted as the same word.
```

Certainly! Below is the completed script with the `clean_line` function implemented to remove punctuation and convert the text to lowercase. Additionally, I've made some minor corrections to ensure the script runs smoothly.

```
```python
file_processor.py - Complete script for
processing text files
import string

def load_file(filepath):
 with open(filepath, 'r') as f:
 return f.readlines()

def clean_line(line):
 # Remove punctuation and make lowercase
 translator = str.maketrans('', '',
string.punctuation)
 clean = line.translate(translator).lower()
 return clean

def count_words(lines):
 word_counts = {}
 for line in lines:
 clean = clean_line(line)
 for word in clean.split():
 word_counts[word] =
word_counts.get(word, 0) + 1
 return word_counts

def main():
 filepath = 'input.txt'
 lines = load_file(filepath)
 counts = count_words(lines)
 for word, count in sorted(counts.items()):
 print(f"{word}: {count}")

if __name__ == '__main__':
 main()
...`
```

### Explanation of Changes:

```

* **`line.translate(str.maketrans('', '',
string.punctuation))`** This is the most efficient
way to remove punctuation in Python.
* `string.punctuation` provides a string
containing all standard punctuation characters.
* `str.maketrans('', '', string.punctuation)`
creates a translation table that maps each punctuation
character to `None`, effectively deleting it. The
first two arguments being empty strings mean that no
characters are replaced; only deleted.
* `line.translate(translation_table)` applies
the translation table to the line.

* **Corrected `filepath` in `main()`** Changed
`filepath = 'input.txt '` to `filepath =
'input.txt'`. The space was causing the file to not be
found. It's important to make sure the file name
matches *exactly*.

* **Clear Comments** Added comments to explain the
purpose of the code.

* **Efficiency** Using `str.maketrans` and
`translate` is generally more efficient for removing
multiple characters (like all punctuation) than
repeated calls to `replace`.

...
1. **`clean_line` Function:**
- `str.maketrans('', '', string.punctuation)`
creates a translation table that maps each
punctuation character to `None`.
- `line.translate(translator)` removes the
punctuation from the line.
- `.lower()` converts the line to lowercase.

2. **Minor Corrections:**
- Removed the extra spaces around the
`input.txt` string in the `filepath` variable.
- Corrected the indentation for the `print`
statement inside the `main` function.

Usage:
- Ensure you have an `input.txt` file in the same
directory as the script.
- Run the script using a Python interpreter:
`python file_processor.py`.

This script will read the file, clean each line,
count the frequency of each word, and print the
results in alphabetical order.

```

Both models here provide similar solutions for completing the code with the Zero-Shot strategy, however, Gemini left behind some minor syntax errors from the original script basis, instead of fixing them like Codestral. We can see, however, that Gemini did fix major syntax errors and provided it in its explanation.

### Prompt:

Could you please complete the following python script so that it cleans each line, removes punctuation, and counts word frequency:

```
file_processor .py - Incomplete script for processing text files
import string
def load_file (filepath) :
 with open (filepath , 'r') as f :
 return f . readlines ()
def clean_line (line) :
 # TODO : Remove punctuation and make lowercase
 pass
def count_words (lines) :
 word_counts = {}
 for line in lines :
 clean = clean_line (line)
 for word in clean . split () :
 word_counts [word] = word_counts . get (word , 0) + 1
 return word_counts
def main () :
 filepath = 'input .txt '
 lines = load_file (filepath)
 counts = count_words (lines)
 for word , count in sorted (counts . items ()) :
 print (f"{ word } : { count }")
if __name__ == '__main__' :
 main ()
```

Explain your thought process step-by-step.

### Strategy:

### Chain of Thought

#### Gemini 2.0 Flash

#### Codestral 25.10

```
```python
# file_processor .py - Incomplete script for
processing text files
import string
def load_file ( filepath ) :
    with open ( filepath , 'r') as f :
        return f . readlines ()
def clean_line ( line ) :
    # TODO : Remove punctuation and make lowercase
    line = line.lower()
    line = line.translate(str.maketrans('', '',
string.punctuation))
    return line
def count_words ( lines ) :
    word_counts = {}
    for line in lines :
        clean = clean_line ( line )
        for word in clean . split () :
            word_counts [ word ] = word_counts . get (
word , 0 ) + 1
    return word_counts
def main () :
    filepath = 'input .txt '
    lines = load_file ( filepath )
    counts = count_words ( lines )
    for word , count in sorted ( counts . items () ) :
        print ( f"{ word } : { count }")
if __name__ == '__main__' :
    main ()
```
```

**\*\*Step-by-step explanation of the `clean\_line` function:\*\***

#### 1. \*\*Lowercase Conversion:\*\*

- `line = line.lower()`: The first step is to convert the entire input `line` to lowercase. This ensures that words like "The" and "the" are treated as the same word when counting frequency, preventing them from being counted separately.

#### 2. \*\*Punctuation Removal:\*\*

Sure! Let's break down the task step-by-step and complete the script.

#### ### Step-by-Step Explanation:

##### 1. \*\*Loading the File:\*\*

- The `load\_file` function is correct. It reads all lines from the file and returns them as a list of strings.

##### 2. \*\*Cleaning the Line:\*\*

- The `clean\_line` function needs to:
- Remove punctuation.
- Convert the text to lowercase.

##### 3. \*\*Counting Word Frequency:\*\*

- The `count\_words` function is mostly correct. It iterates through the lines, cleans each line, splits it into words, and counts the frequency of each word.

##### 4. \*\*Main Function:\*\*

- This function sets the file path, loads the lines, counts the word frequencies, and prints the results.

#### ### Complete Script:

Here is the completed script:

```
```python
# file_processor.py - Complete script for
processing text files
import string

def load_file(filepath):
    with open(filepath, 'r') as f:
        return f.readlines()

def clean_line(line):
    # Remove punctuation and make lowercase
    translator = str.maketrans('', '',
string.punctuation)

    line = line.lower()
    line = line.translate(translator)
    return line

def count_words(lines):
    word_counts = {}
    for line in lines:
        clean = clean_line(line)
        for word in clean.split():
            word_counts[word] = word_counts.get(word, 0) + 1
    return word_counts

def main():
    filepath = 'input.txt'
    lines = load_file(filepath)
    counts = count_words(lines)
    for word, count in sorted(counts.items()):
        print(f"{word} : {count}")

if __name__ == '__main__':
    main()
```
```

```

- `line = line.translate(str.maketrans('', '',
string.punctuation))`: This line is the core of
punctuation removal. Let's break it down:
- `string.punctuation`: This is a string constant
from the `string` module that contains all standard
punctuation characters (e.g.,
`!"#$%&'()*+,-./:;<=>?@[\\]^_`{|}~`).
- `str.maketrans('', '', string.punctuation)`:
This creates a translation table. The first two
arguments would typically be used to replace
characters with other characters. Here, they are empty
strings, meaning no replacement is performed. The
third argument, `string.punctuation`, tells
`maketrans` that we want to *delete* all characters
found in `string.punctuation`.
- `line.translate(...)`: The `translate` method
then applies this translation table to the input
`line`. Effectively, it removes all punctuation
characters from the string.

3. **Return the Cleaned Line:**
- `return line`: The function returns the modified
`line`, which is now lowercase and devoid of any
punctuation.

...

cleaned_line =
line.translate(translator).lower()
return cleaned_line

def count_words(lines):
 word_counts = {}
 for line in lines:
 clean = clean_line(line)
 for word in clean.split():
 word_counts[word] =
word_counts.get(word, 0) + 1
 return word_counts

def main():
 filepath = 'input.txt'
 lines = load_file(filepath)
 counts = count_words(lines)
 for word, count in sorted(counts.items()):
 print(f"{word}: {count}")

if __name__ == '__main__':
 main()
'''

Explanation of Changes:

1. **`clean_line` Function:**
- We use `str.maketrans` to create a translation
table that maps each punctuation character to
`None`.
- `line.translate(translator)` removes the
punctuation.
- `lower()` converts the line to lowercase.

2. **Main Function:**
- Corrected the file path to `input.txt`
without spaces.

...

```

As discussed in the previous task, once again the Chain of Thought strategy seems to have a lesser effect on the larger and more complex scripts. This could be because the Model's were reaching their token limit (1024), however the step-by-step was less thorough and unique for both Models and was very similar to the explanations given in the Zero-Shot strategy. Both of the coding answers given were also the same as the previous prompt.

# Conclusion

## Prompting Strategy Comparison

I think each kind of task posed unique problems that only certain strategies could properly solve. In order to better compare the strategies, I will divide the comparisons in terms of **NLP** tasks and **Code Generation** tasks.

### **NLP**

These tasks typically included summarising\describing code or describing bugs/errors. When asked to identify or describe bugs/errors, no prompting strategy was able to keep a model on task. A lot of information was cut out from tasks like Task 7, where even Zero-Shot prompting strategies would elicit Chain of Thought level description. In this same regard all prompting strategies were equally effective on these tasks. The Prompt Chaining strategy was especially egregious due to the inability for models to remain on task. This caused a lot of models to repeat themselves when the second clarifying prompt was presented to them. Although I do not know how to prove this, it felt as if the system or rules prompt provided by the designers of each model caused this behaviour in the model when identifying bugs. I think each prompt for this particular type of task would need to be edited with further length and scope limitations to more effectively and concisely complete the tasks.

On the other hand, summarisation tasks varied quite a lot depending on the strategy used. As one example, looking at Task 17 and observing Gemini's responses, we can see that Few-Shot prompting is very effective at eliciting a concise response from the model, avoiding any off-task explanations. Looking at other strategies like Zero-Shot, Role-Play, and Self-Consistency, (using Task 11 and Task 19 as examples), they clearly have almost no effect on the difference in responses as a whole. Chain of Thought and Prompt Chaining strategies (using Task 18) were both extremely helpful in adding information, however, they would also repeat themselves often.

Overall, I would say that Few-Shot prompting showed itself as the most effective technique for these NLP tasks, however Chain of Thought and Prompt Chaining strategies could be utilised if a more verbose answer is wanted.

### **Code Generation**

These tasks are much more interesting to look at, especially when comparing correctness alongside reasoning from the Model. Responses to each code generation task were extremely varied (especially between models) for the same strategies, however, some trends became apparent:

Few-shot learning continued to be the most effective way in which to obtain concise responses. For example, ChatGPT-4o's response in Task 14 given the Few-Shot learning was extremely concise, and it even gave an answer in the same coding style as the one given in the shot.

Zero-Shot, Self-Consistency, and Role-Play strategies continued to not cause a lot of change. When these three strategies were being compared, they were almost all comparable. Taking a look at Task 8 we see almost no change. However, depending on what role the Model was given, the Role-Play strategy might cause some change. In that same task, the Model is given the role of a "helpful coding assistant" and both model responses increase the verbosity of their code, however, when given another more specific (but sillier) role, "master hacker," it almost seems as if the model ignores it.

Prompt Chaining prompts struggled through this task type as well, for a similar reason. Even when decreasing the scope significantly of the initial prompt, the Model would still answer the original question. In Task 21, we do see an example of extra information being added after the clarifying prompt in the coding solution, specifically adding the logging module.

Finally, I was able to notice a more peculiar behaviour of the Chain of Thought strategy. For most models and for most short tasks, a model would use the phrase "Understanding the ..." or "Understand the ..." (eg. Task 14) but not in response to other strategies. This is not true of longer tasks, like 19 and onwards, where the models then top using the Chain of Thought.

## Model Comparison

The models each have unique characteristics that become more apparent in their responses to the different strategies.

I would say the best model for every task by far was Gemini 2.0 Flash. It would remain on task the most and even sometimes complete the task perfectly, like in Task 17. It was most responsive to the Few-Shot prompting strategy and always put the code answer first in coding tasks.

ChatGPT-4o was almost equal to Gemini, however, always was able to add in extra off-task information or included repetition. One example is in Task 14 when ChatGPT-4o gives a very accurate response, however provides two answers rather than a single best one.

Llama-4-Maverick and Codestral 25.10 both struggle with verbosity, especially for bug identification tasks and code generation tasks. Their answers, no matter the strategy, were often repetitive and off-task, however, they were always descriptive. Codestral additionally seems more tuned towards junior programmers; looking at Task 20, Codestral writes out all the loops rather than using the built-in python functions to make code shorter like Gemini does.

Overall, looking at both the above sections, it becomes clear that I think that the most effective manner in which I should complete the tasks above, were I to make a pipeline to do so, would be to design Few-Shot prompts and feed them to Gemini.