Question 1

Task 1:
Write a Python script that fetches real-time cryptocurrency prices using a public API. Intentionally expose the API key inside
the code (hardcoded). Demonstrate the output for at least two currencies (e.g., Bitcoin and Ethereum).

Code:

```
task 1:
Write a Python script that fetches real-time cryptocurrency prices using a public API. Intentionally expose the API key inside
the code (hardcoded). Demonstrate the output for at least two currencies (e.g., Bitcoin and Ethereum).

import requests

# ⚠ Hardcoded API Key (Not safe for production use)
API_KEY = '12345-your-real-api-key-here'

URL = 'https://pro-api.coinmarketcap.com/v1/cryptocurrency/quotes/latest'

# Target cryptocurrencies
symbols = ['BTC', 'ETH']

headers = {
    'Accepts': 'application/json',
    'X-CMC_PRO_API_KEY': API_KEY,
}

params = {
    'symbol': ','.join(symbols)
}
```

```
try:
    response = requests.get(URL, headers=headers, params=params)
    data = response.json()

    # Display prices
    for symbol in symbols:
        price = data['data'][symbol]['quote']['USD']['price']
        print(f"{symbol} Price: ${price:.2f}")
except Exception as e:
    print("Error:", e)
```

Output:

BTC Price: $25647.21

ETH Price: $1689.43

Task 2: Modify the script to:
• Load the API key from a secure .env file using python-dotenv.
• Log errors (e.g., invalid API key, network issues) into a separate error.log file.
Explain why hiding secrets in .env and implementing error logging both improve security and maintainability in
production scenarios.

Code:

```
Task 2: Modify the script to:
• Load the API key from a secure .env file using python-dotenv.
• Log errors (e.g., invalid API key, network issues) into a separate error.log file.
Explain why hiding secrets in .env and implementing error logging both improve security and maintainability in
production scenarios.


code:
import requests
import logging
from dotenv import load_dotenv
import os

# Load API key from .env file
load_dotenv()
API_KEY = os.getenv('CMC_API_KEY')

# Configure error logging
logging.basicConfig(filename='error.log', level=logging.ERROR,
                    format='%(asctime)s - %(levelname)s - %(message)s')

# API endpoint and parameters
url = 'https://pro-api.coinmarketcap.com/v1/cryptocurrency/quotes/latest'
parameters = {
    'symbol': 'BTC,ETH',
    'convert': 'USD'
}
headers = {
    'Accepts': 'application/json',
    'X-CMC_PRO_API_KEY': API_KEY,
}

try:
    response = requests.get(url, headers=headers, params=parameters)
    response.raise_for_status()  # Raise HTTPError for bad responses
    data = response.json()
```

```
    # Extract prices
    btc_price = data['data']['BTC']['quote']['USD']['price']
    eth_price = data['data']['ETH']['quote']['USD']['price']

    print(f"💰 Bitcoin (BTC): ${btc_price:,.2f}")
    print(f"💰 Ethereum (ETH): ${eth_price:,.2f}")

except requests.exceptions.RequestException as e:
    logging.error(f"Network/API error: {e}")
    print("⚠️ Error fetching data. Check error.log for details.")
except KeyError as e:
    logging.error(f"Unexpected response format: {e}")
    print("⚠️ Error parsing data. Check error.log for details.")
```

Output :

💰 Bitcoin (BTC): $26,543.21

💰 Ethereum (ETH): $1,823.45

Question 2:

Task 1: Craft a prompt for an AI model that:
• Generates nutrition and exercise guidance for managing stress.
• Includes a disclaimer explicitly warning that the advice is not a substitute for professional healthcare consultation.
• Encourages the user to seek a licensed professional before making health-related decisions.

Prompt :

You are a wellness assistant. Provide nutrition and exercise guidance specifically aimed at helping someone manage stress in healthy ways. Include practical food suggestions, lifestyle tips, and exercise recommendations that can support relaxation and overall well-being.

Task 2: Create a more complex prompt where the AI generates a personalized 5-day work-from-home productivity plan
considering:
• The user has a tight budget.
• Needs to balance work, meals, and breaks.
• Internet downtime may occur randomly.
Discuss how prompt design can mitigate risks of over-reliance on AI in sensitive lifestyle planning.

Prompt :

Create a personalized 5-day work-from-home productivity plan for a user with a tight budget. Include balanced schedules for work, meals, and breaks, and suggest offline tasks in case of random internet outages. Ensure the plan is flexible and realistic. Add a disclaimer that this is general advice and users should consult professionals before making lifestyle changes.