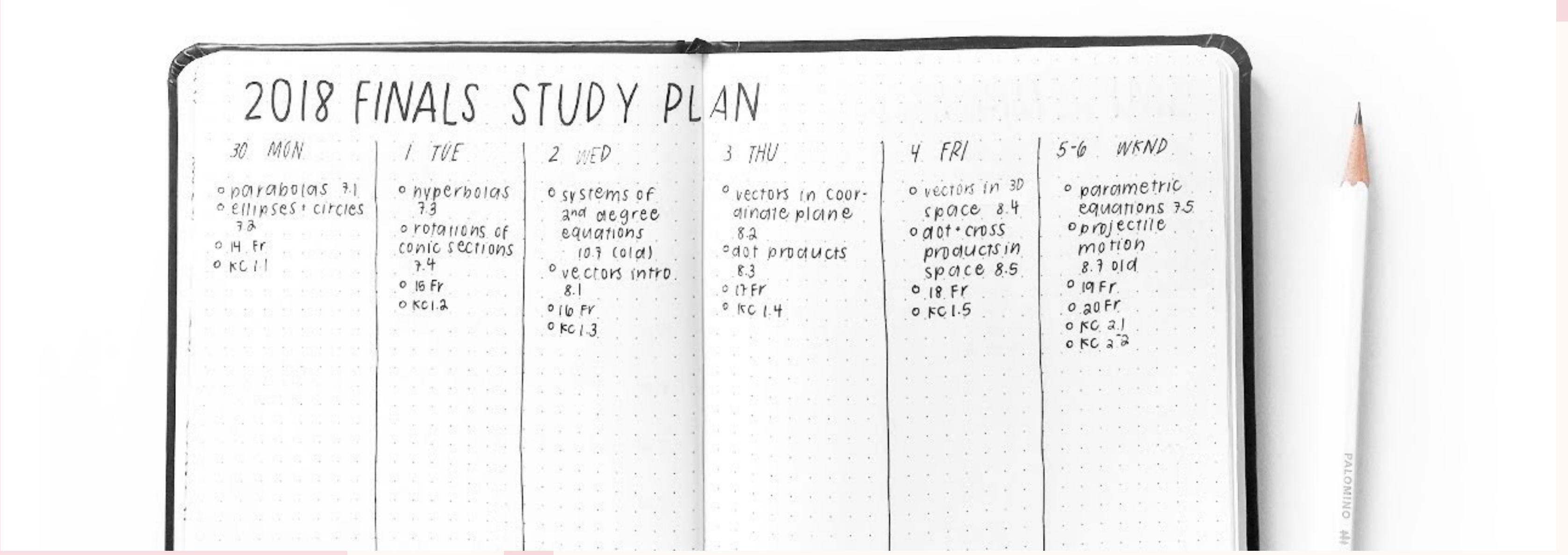
Prompt Engineering Project

Roshny Titus

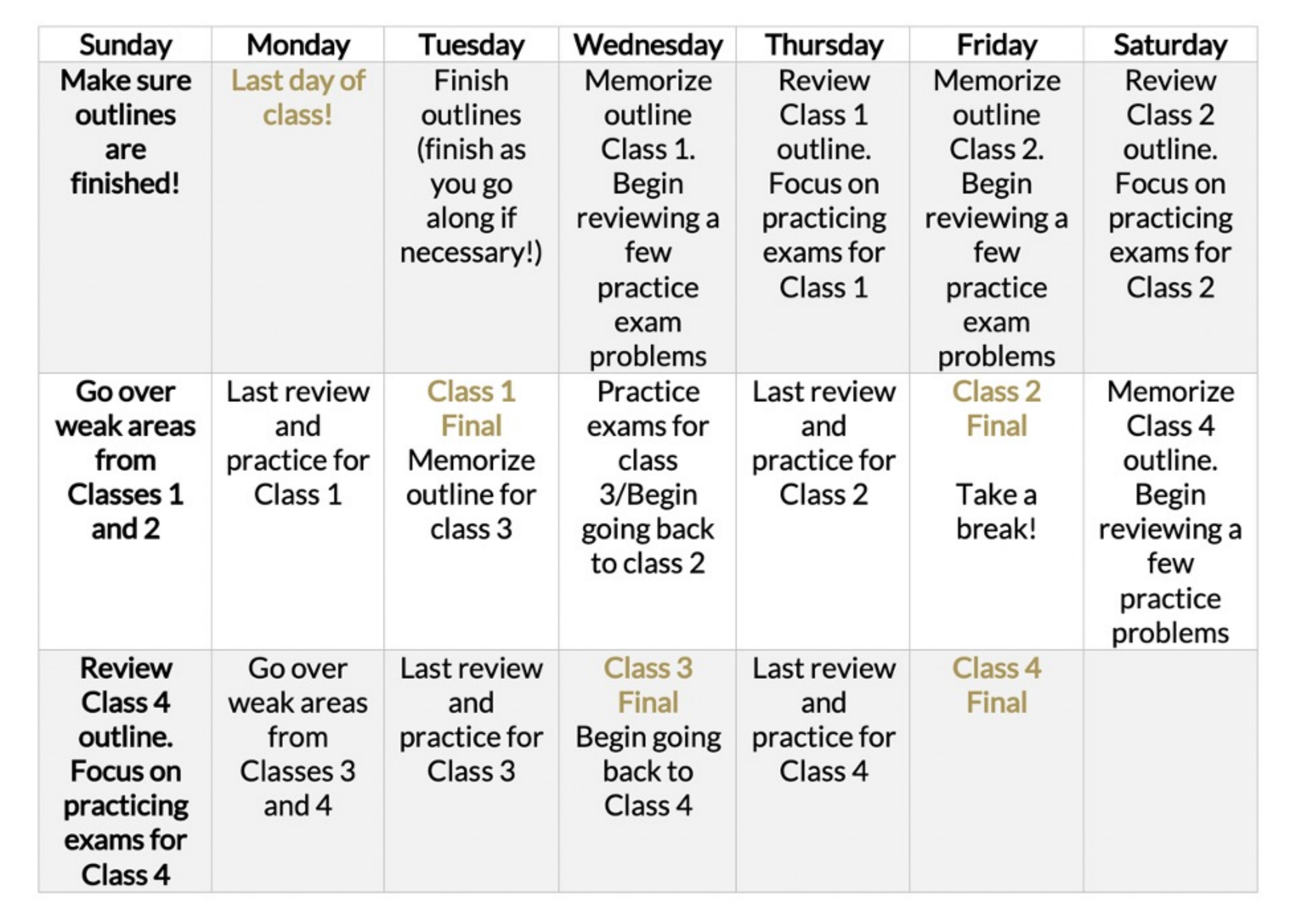
Goal

Use Chat GPT to make me a detailed exam study schedule



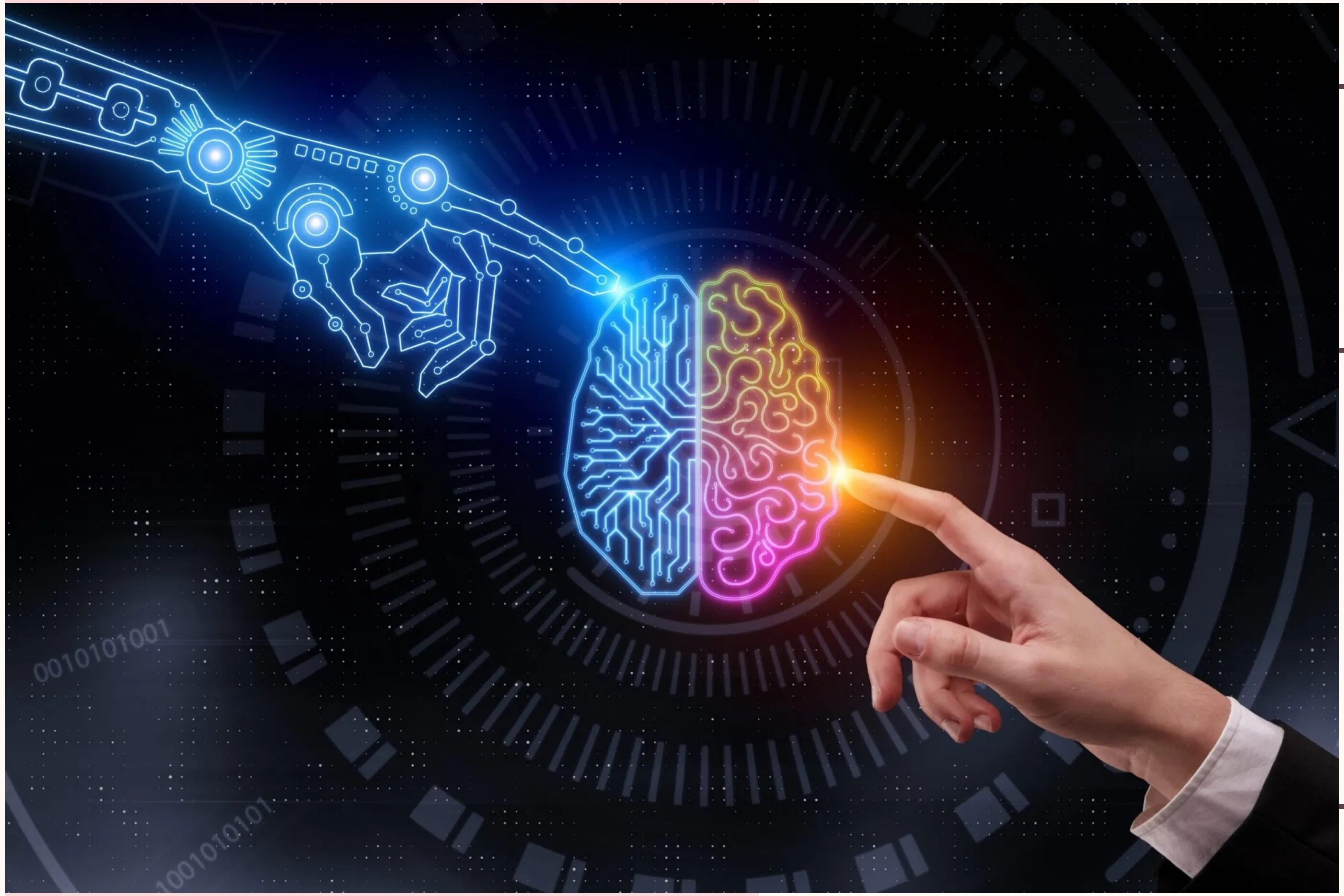
2018 FINALS STUDY PLAN

30 MON	1 TUE	2 WED	3 THU	4 FRI	5-6 WKND
○ parabolas 7.1 ○ ellipses + circles 7.2 ○ 14 Fr ○ KC 1.1	○ hyperbolas 7.3 ○ rotations of conic sections 7.4 ○ 15 Fr ○ KC 1.2	○ systems of 2nd degree equations 10.7 (old) ○ vectors intro 8.1 ○ 16 Fr ○ KC 1.3	○ vectors in coordinate plane 8.2 ○ dot products 8.3 ○ 17 Fr ○ KC 1.4	○ vectors in 3D space 8.4 ○ dot + cross products in space 8.5 ○ 18 Fr ○ KC 1.5	○ parametric equations 7.5 ○ projectile motion 8.7 old ○ 19 Fr ○ 20 Fr ○ KC 2.1 ○ KC 2.2



Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Make sure outlines are finished!	Last day of class!	Finish outlines (finish as you go along if necessary!)	Memorize outline Class 1. Begin reviewing a few practice exam problems	Review Class 1 outline. Focus on practicing exams for Class 1	Memorize outline Class 2. Begin reviewing a few practice exam problems	Review Class 2 outline. Focus on practicing exams for Class 2
Go over weak areas from Classes 1 and 2	Last review and practice for Class 1	Class 1 Final Memorize outline for class 3	Practice exams for class 3/Begin going back to class 2	Last review and practice for Class 2	Class 2 Final Take a break!	Memorize Class 4 outline. Begin reviewing a few practice problems
Review Class 4 outline. Focus on practicing exams for Class 4	Go over weak areas from Classes 3 and 4	Last review and practice for Class 3	Class 3 Final Begin going back to Class 4	Last review and practice for Class 4	Class 4 Final	

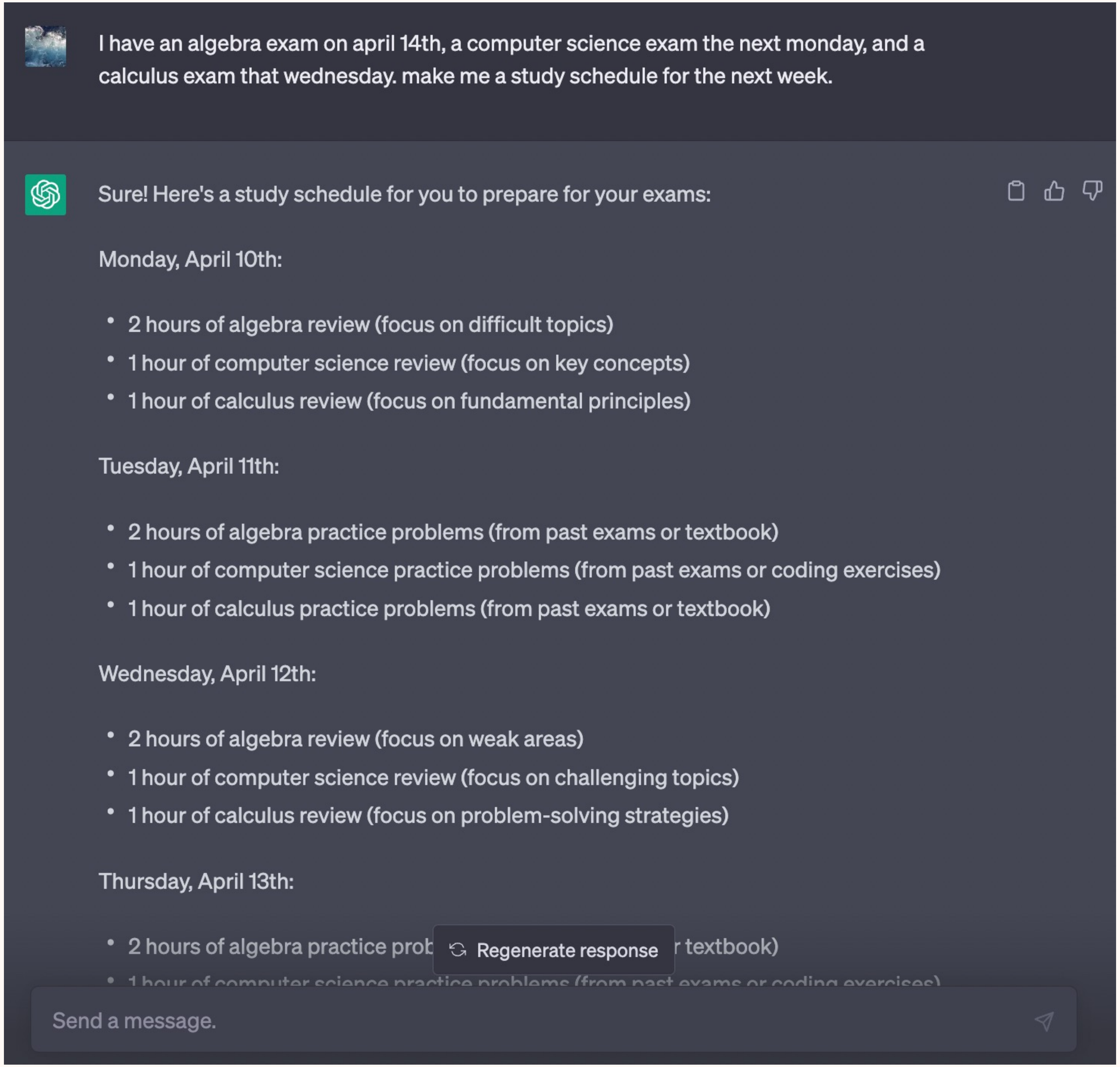
Process



- Experiment with Chat GPT prompts
- Experiment with using Chat GPT to generate code
- Use Chat GPT to generate code for a Python program that reads exam info from the user and generates a study schedule with Chat GPT API

Process

Experiment with Chat GPT prompts



I have an algebra exam on april 14th, a computer science exam the next monday, and a calculus exam that wednesday. make me a study schedule for the next week.

Sure! Here's a study schedule for you to prepare for your exams:

Monday, April 10th:

- 2 hours of algebra review (focus on difficult topics)
- 1 hour of computer science review (focus on key concepts)
- 1 hour of calculus review (focus on fundamental principles)

Tuesday, April 11th:

- 2 hours of algebra practice problems (from past exams or textbook)
- 1 hour of computer science practice problems (from past exams or coding exercises)
- 1 hour of calculus practice problems (from past exams or textbook)

Wednesday, April 12th:

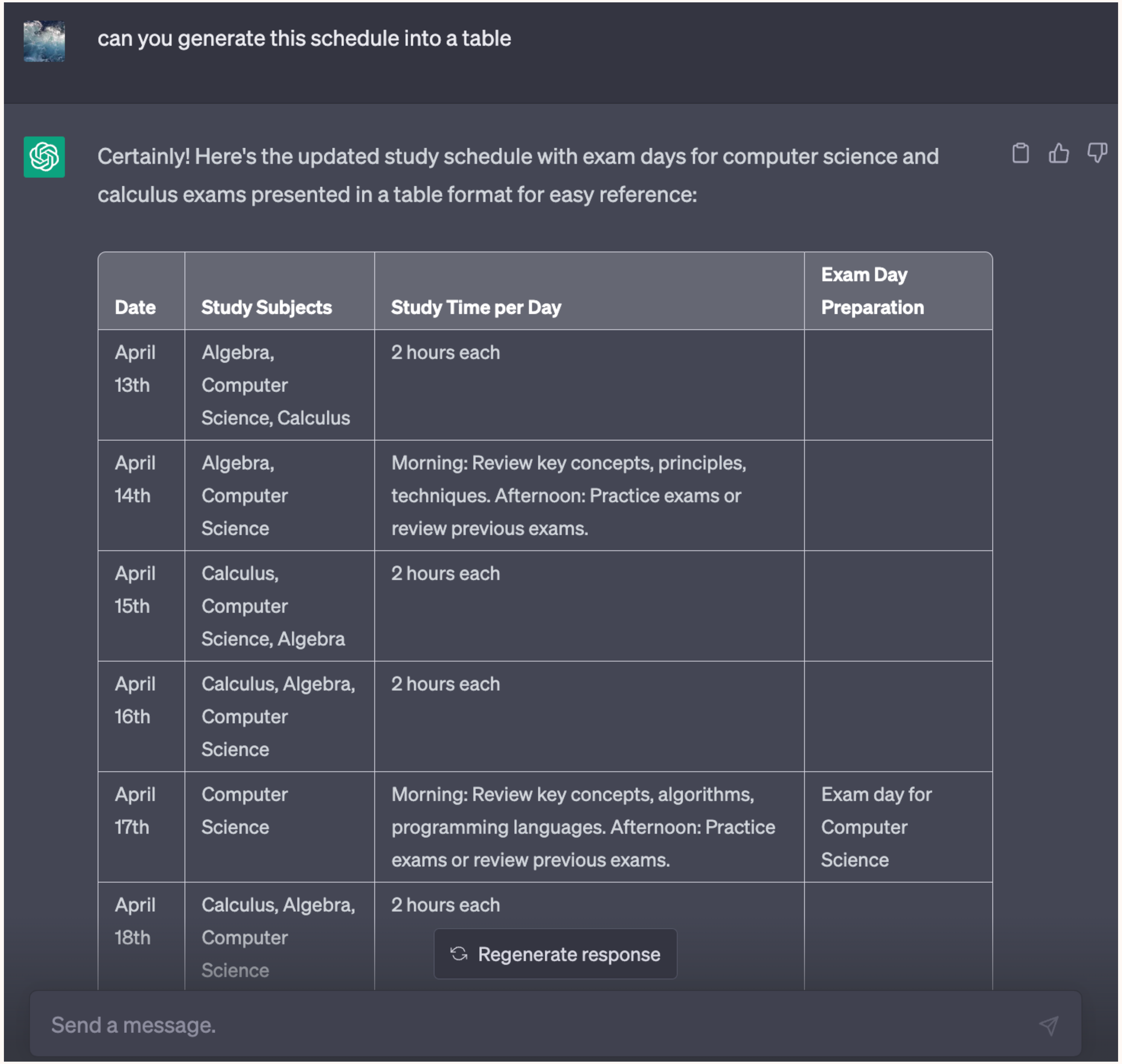
- 2 hours of algebra review (focus on weak areas)
- 1 hour of computer science review (focus on challenging topics)
- 1 hour of calculus review (focus on problem-solving strategies)

Thursday, April 13th:

- 2 hours of algebra practice prob r textbook)
- 1 hour of computer science practice problems (from past exams or coding exercises)

Process

Experiment with Chat GPT prompts



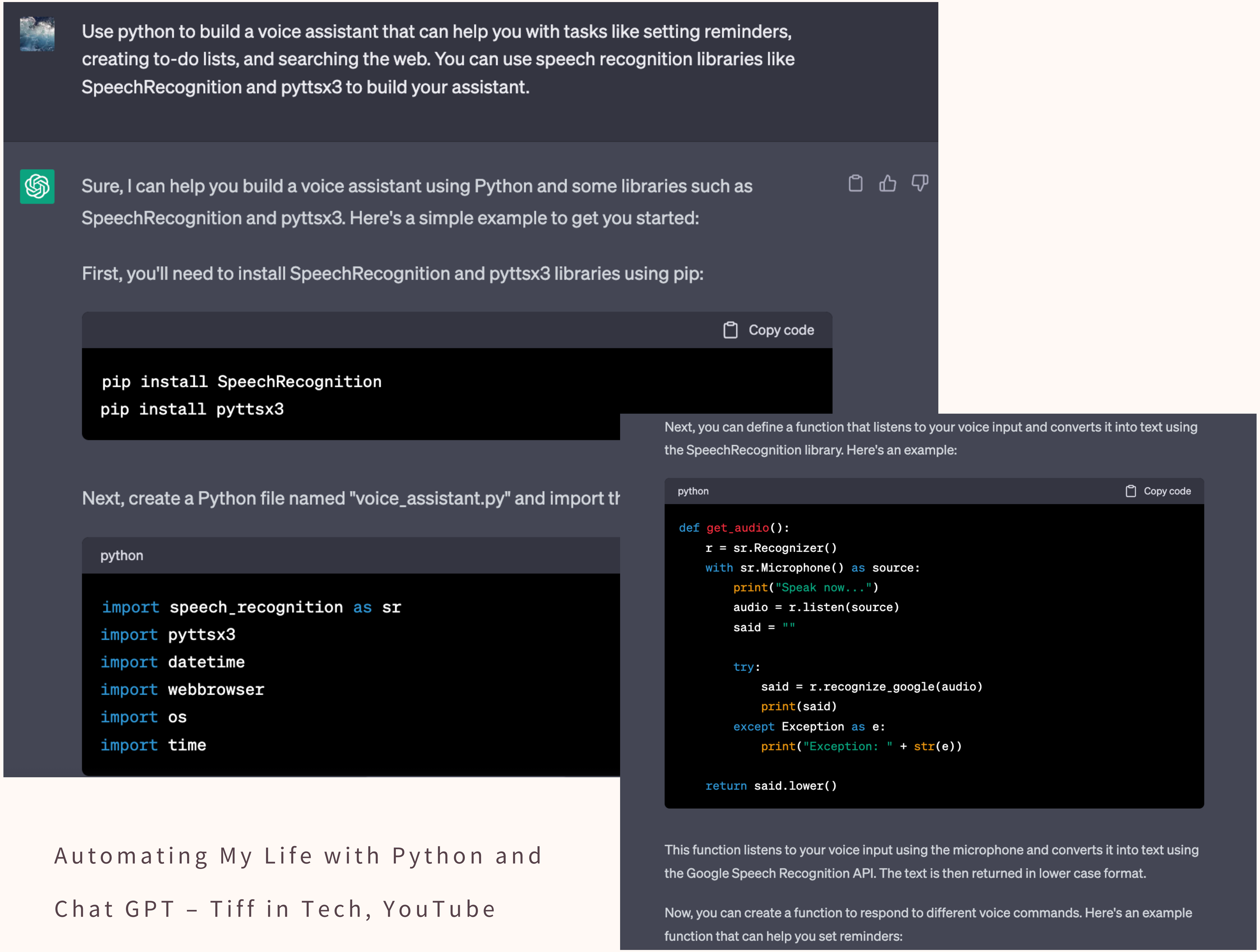
can you generate this schedule into a table

Certainly! Here's the updated study schedule with exam days for computer science and calculus exams presented in a table format for easy reference:

Date	Study Subjects	Study Time per Day	Exam Day Preparation
April 13th	Algebra, Computer Science, Calculus	2 hours each	
April 14th	Algebra, Computer Science	Morning: Review key concepts, principles, techniques. Afternoon: Practice exams or review previous exams.	
April 15th	Calculus, Computer Science, Algebra	2 hours each	
April 16th	Calculus, Algebra, Computer Science	2 hours each	
April 17th	Computer Science	Morning: Review key concepts, algorithms, programming languages. Afternoon: Practice exams or review previous exams.	Exam day for Computer Science
April 18th	Calculus, Algebra, Computer Science	2 hours each	

Process

Experiment with using Chat GPT to generate code



Use python to build a voice assistant that can help you with tasks like setting reminders, creating to-do lists, and searching the web. You can use speech recognition libraries like SpeechRecognition and pyttsx3 to build your assistant.

Sure, I can help you build a voice assistant using Python and some libraries such as SpeechRecognition and pyttsx3. Here's a simple example to get you started:

First, you'll need to install SpeechRecognition and pyttsx3 libraries using pip:

```
pip install SpeechRecognition
pip install pyttsx3
```

Next, create a Python file named "voice_assistant.py" and import t

```python
import speech_recognition as sr
import pyttsx3
import datetime
import webbrowser
import os
import time
```

Next, you can define a function that listens to your voice input and converts it into text using the SpeechRecognition library. Here's an example:

```python
def get_audio():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Speak now...")
        audio = r.listen(source)
        said = ""

        try:
            said = r.recognize_google(audio)
            print(said)
        except Exception as e:
            print("Exception: " + str(e))

    return said.lower()
```

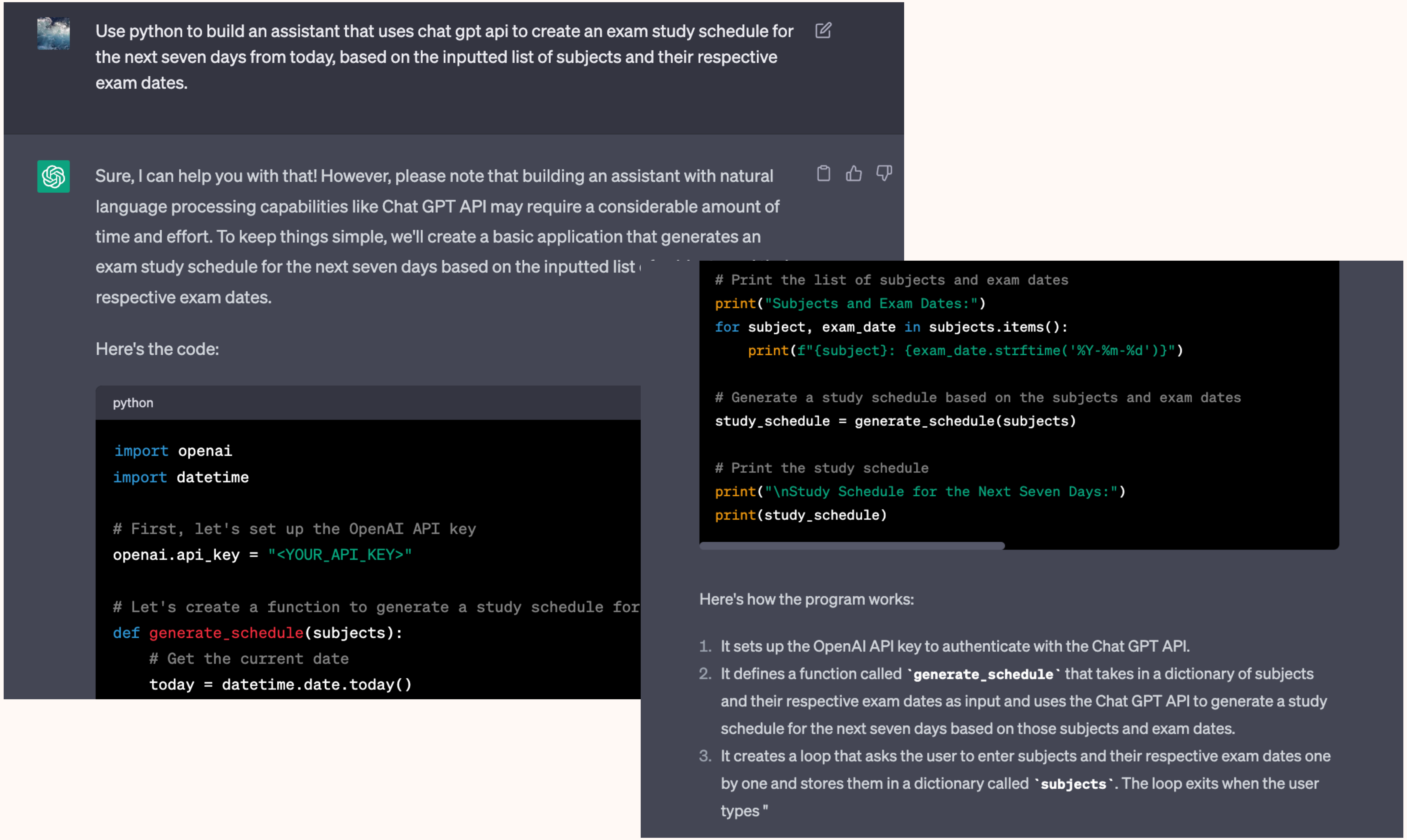
This function listens to your voice input using the microphone and converts it into text using the Google Speech Recognition API. The text is then returned in lower case format.

Now, you can create a function to respond to different voice commands. Here's an example function that can help you set reminders:

Automating My Life with Python and Chat GPT – Tiff in Tech, YouTube

Process

Use Chat GPT to generate code for a Python program that reads exam info from the user and generates a study schedule with Chat GPT API



Use python to build an assistant that uses chat gpt api to create an exam study schedule for the next seven days from today, based on the inputted list of subjects and their respective exam dates.

Sure, I can help you with that! However, please note that building an assistant with natural language processing capabilities like Chat GPT API may require a considerable amount of time and effort. To keep things simple, we'll create a basic application that generates an exam study schedule for the next seven days based on the inputted list

respective exam dates.

Here's the code:

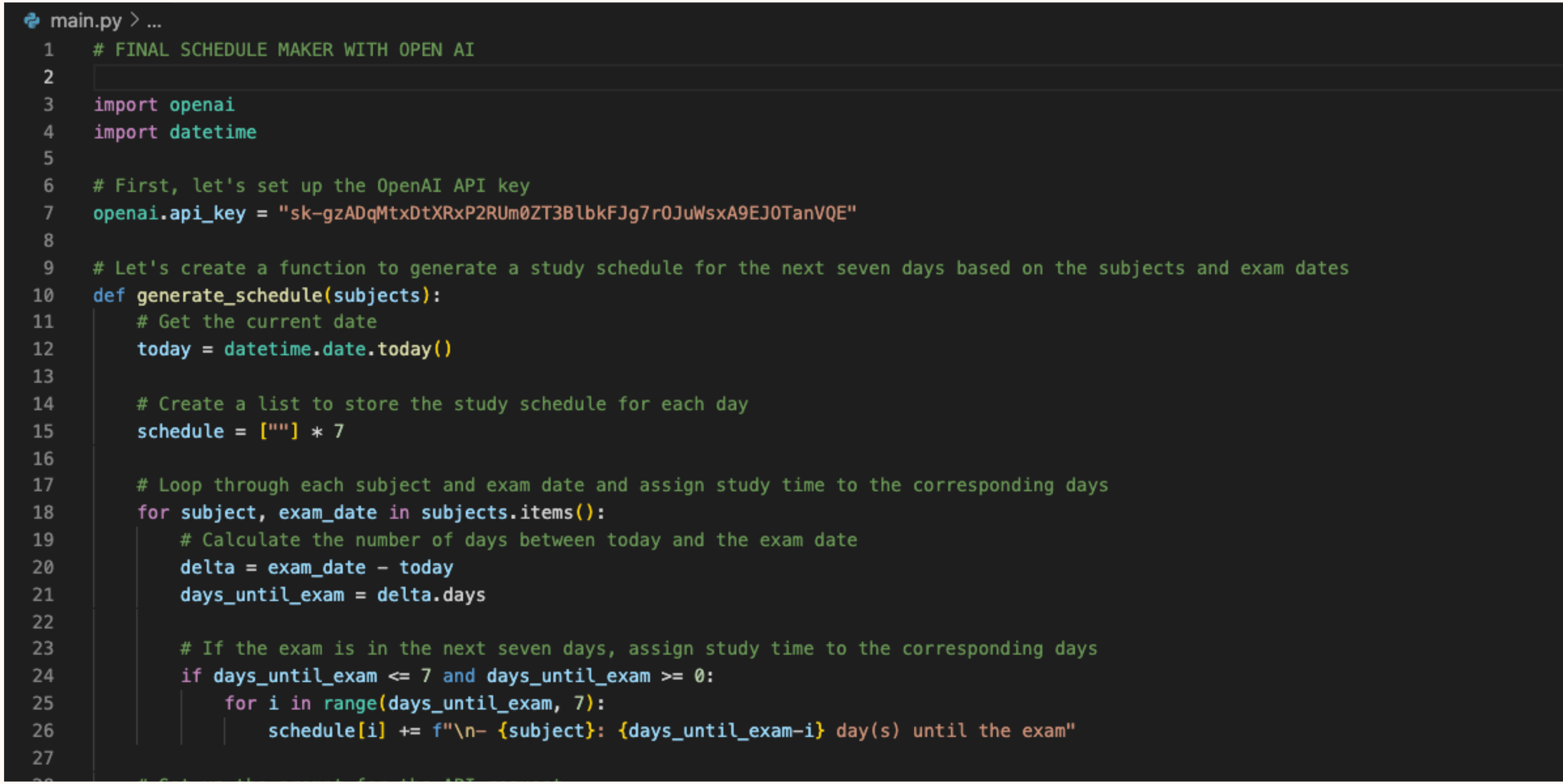
```python
import openai
import datetime

# First, let's set up the OpenAI API key
openai.api_key = "<YOUR_API_KEY>"

# Let's create a function to generate a study schedule for
def generate_schedule(subjects):
    # Get the current date
    today = datetime.date.today()
```

```python
# Print the list of subjects and exam dates
print("Subjects and Exam Dates:")
for subject, exam_date in subjects.items():
    print(f"{subject}: {exam_date.strftime('%Y-%m-%d')}")

# Generate a study schedule based on the subjects and exam dates
study_schedule = generate_schedule(subjects)

# Print the study schedule
print("\nStudy Schedule for the Next Seven Days:")
print(study_schedule)
```

Here's how the program works:

1. It sets up the OpenAI API key to authenticate with the Chat GPT API.
2. It defines a function called `generate_schedule` that takes in a dictionary of subjects and their respective exam dates as input and uses the Chat GPT API to generate a study schedule for the next seven days based on those subjects and exam dates.
3. It creates a loop that asks the user to enter subjects and their respective exam dates one by one and stores them in a dictionary called `subjects`. The loop exits when the user types "

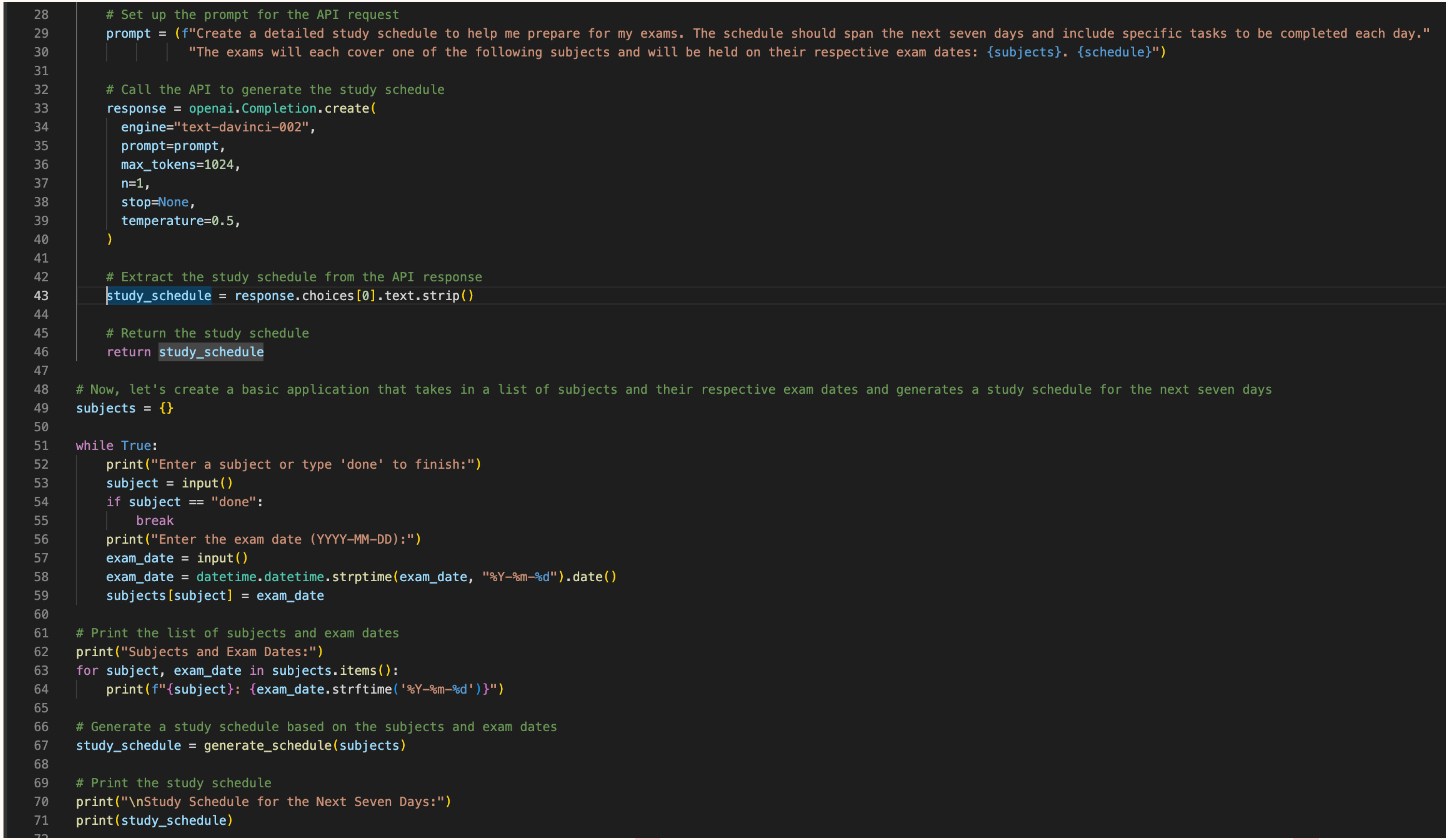
Results



main.py > ...

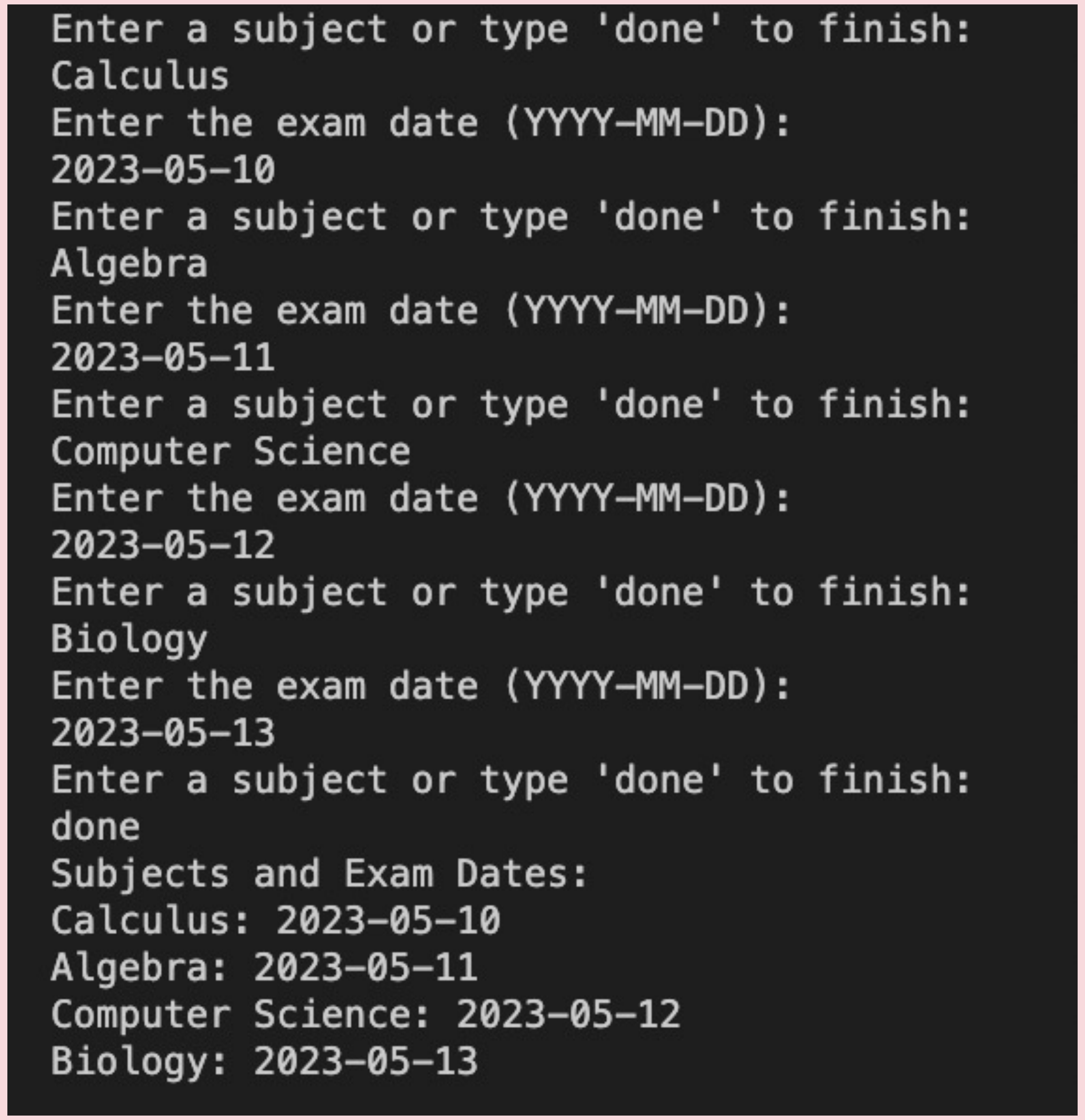
```python
# FINAL SCHEDULE MAKER WITH OPEN AI

import openai
import datetime

# First, let's set up the OpenAI API key
openai.api_key = "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"

# Let's create a function to generate a study schedule for the next seven days based on the subjects and exam dates
def generate_schedule(subjects):
    # Get the current date
    today = datetime.date.today()

    # Create a list to store the study schedule for each day
    schedule = [""] * 7

    # Loop through each subject and exam date and assign study time to the corresponding days
    for subject, exam_date in subjects.items():
        # Calculate the number of days between today and the exam date
        delta = exam_date - today
        days_until_exam = delta.days

        # If the exam is in the next seven days, assign study time to the corresponding days
        if days_until_exam <= 7 and days_until_exam >= 0:
            for i in range(days_until_exam, 7):
                schedule[i] += f"\n- {subject}: {days_until_exam-i} day(s) until the exam"

```

Results

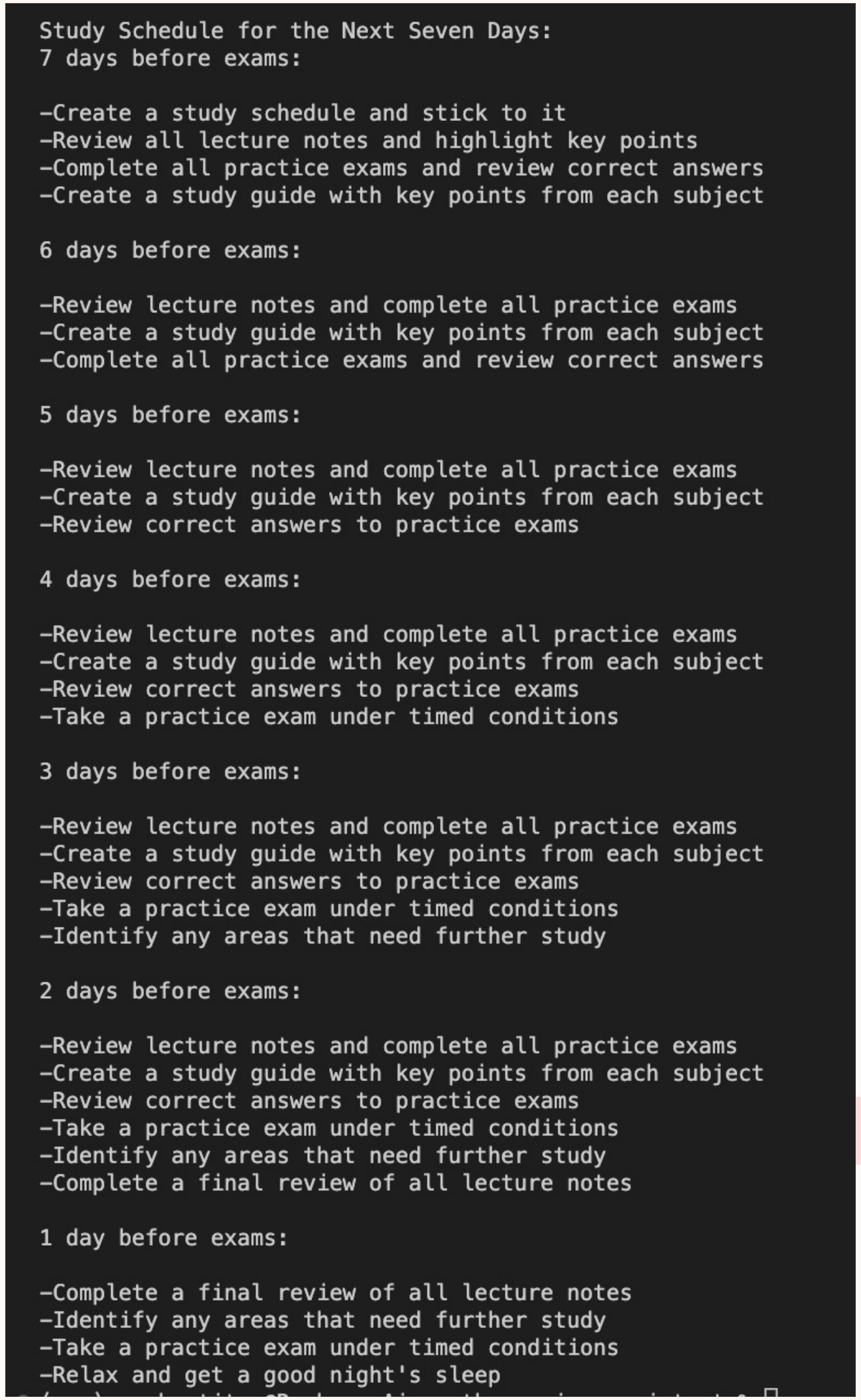


```
    # Set up the prompt for the API request
    prompt = (f"Create a detailed study schedule to help me prepare for my exams. The schedule should span the next seven days and include specific tasks to be completed each day."
               "The exams will each cover one of the following subjects and will be held on their respective exam dates: {subjects}. {schedule}")

    # Call the API to generate the study schedule
    response = openai.Completion.create(
      engine="text-davinci-002",
      prompt=prompt,
      max_tokens=1024,
      n=1,
      stop=None,
      temperature=0.5,
    )

    # Extract the study schedule from the API response
    study_schedule = response.choices[0].text.strip()

    # Return the study schedule
    return study_schedule

# Now, let's create a basic application that takes in a list of subjects and their respective exam dates and generates a study schedule for the next seven days
subjects = {}

while True:
    print("Enter a subject or type 'done' to finish:")
    subject = input()
    if subject == "done":
        break
    print("Enter the exam date (YYYY-MM-DD):")
    exam_date = input()
    exam_date = datetime.datetime.strptime(exam_date, "%Y-%m-%d").date()
    subjects[subject] = exam_date

# Print the list of subjects and exam dates
print("Subjects and Exam Dates:")
for subject, exam_date in subjects.items():
    print(f"{subject}: {exam_date.strftime('%Y-%m-%d')}")

# Generate a study schedule based on the subjects and exam dates
study_schedule = generate_schedule(subjects)

# Print the study schedule
print("\nStudy Schedule for the Next Seven Days:")
print(study_schedule)
```

Results

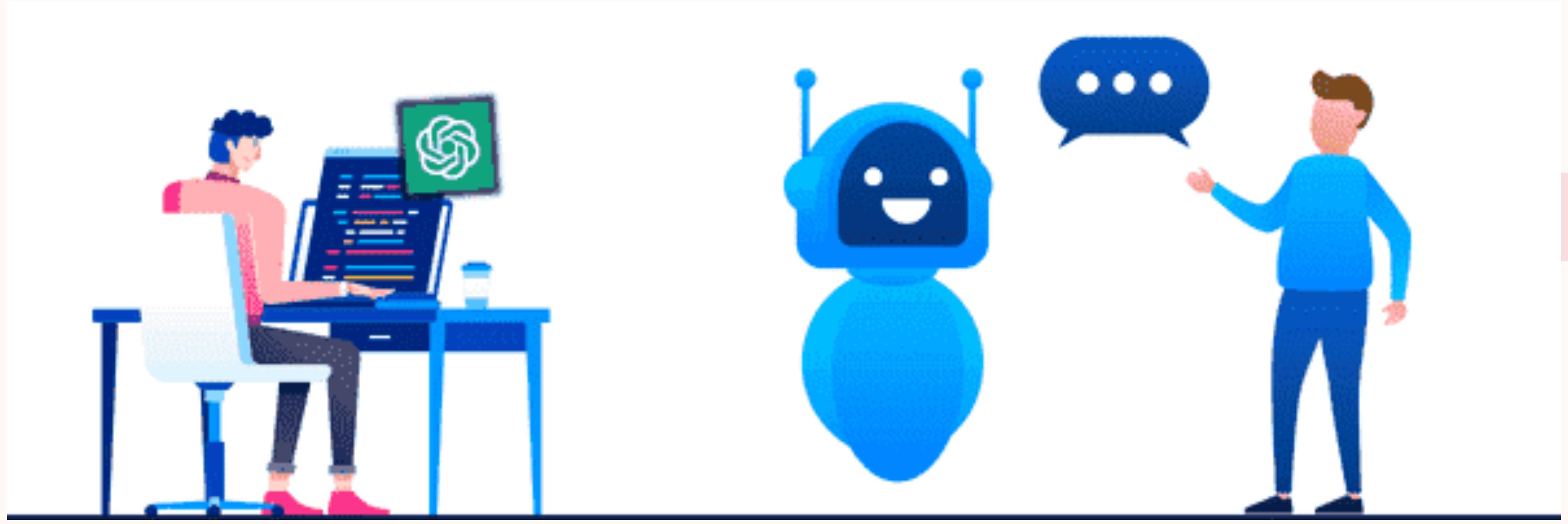


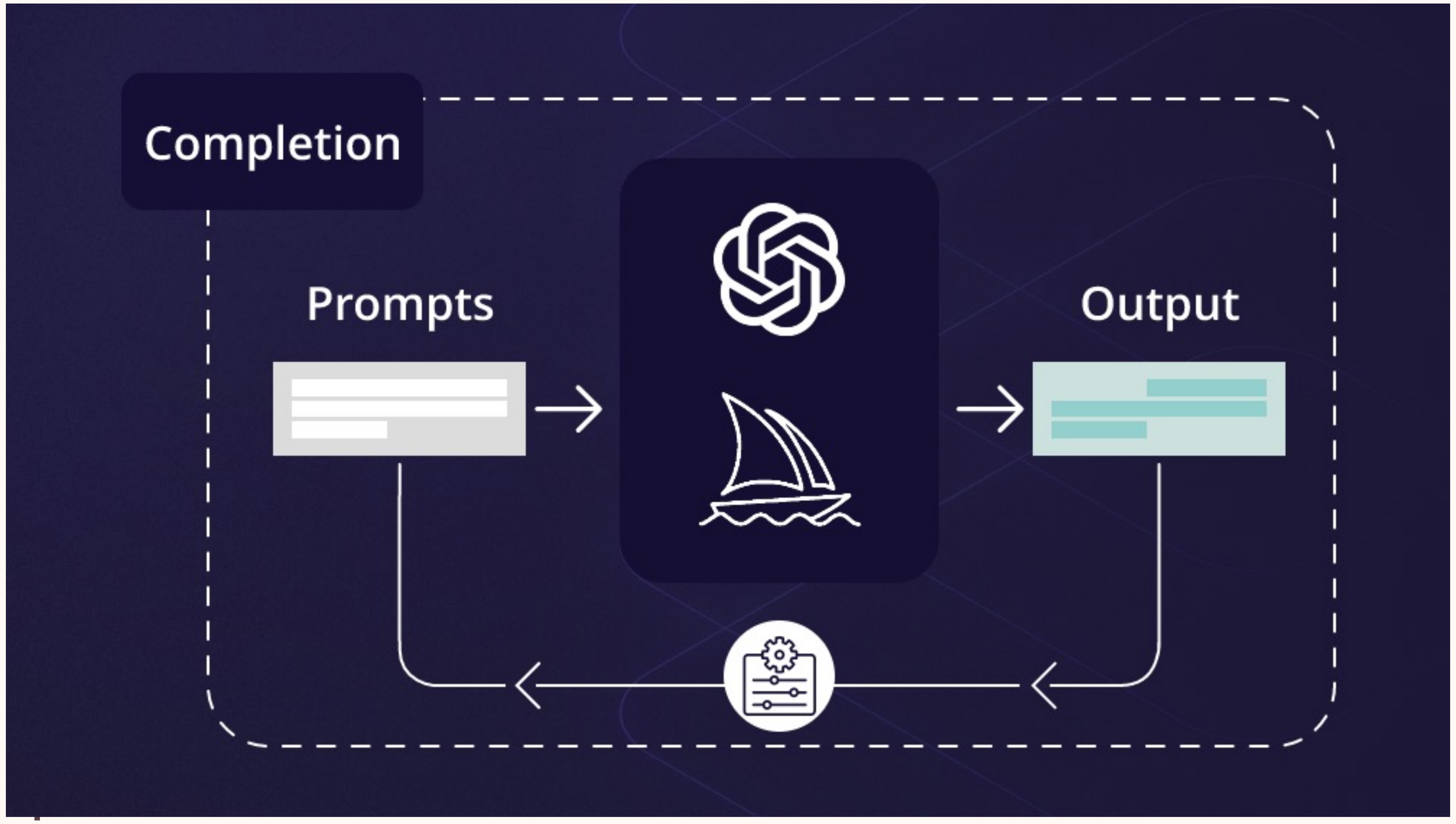
```
Enter a subject or type 'done' to finish:
Calculus
Enter the exam date (YYYY-MM-DD):
2023-05-10
Enter a subject or type 'done' to finish:
Algebra
Enter the exam date (YYYY-MM-DD):
2023-05-11
Enter a subject or type 'done' to finish:
Computer Science
Enter the exam date (YYYY-MM-DD):
2023-05-12
Enter a subject or type 'done' to finish:
Biology
Enter the exam date (YYYY-MM-DD):
2023-05-13
Enter a subject or type 'done' to finish:
done
Subjects and Exam Dates:
Calculus: 2023-05-10
Algebra: 2023-05-11
Computer Science: 2023-05-12
Biology: 2023-05-13
```



```
Study Schedule for the Next Seven Days:
7 days before exams:

-Create a study schedule and stick to it
-Review all lecture notes and highlight key points
-Complete all practice exams and review correct answers
-Create a study guide with key points from each subject

6 days before exams:

-Review lecture notes and complete all practice exams
-Create a study guide with key points from each subject
-Complete all practice exams and review correct answers

5 days before exams:

-Review lecture notes and complete all practice exams
-Create a study guide with key points from each subject
-Review correct answers to practice exams

4 days before exams:

-Review lecture notes and complete all practice exams
-Create a study guide with key points from each subject
-Review correct answers to practice exams
-Take a practice exam under timed conditions

3 days before exams:

-Review lecture notes and complete all practice exams
-Create a study guide with key points from each subject
-Review correct answers to practice exams
-Take a practice exam under timed conditions
-Identify any areas that need further study

2 days before exams:

-Review lecture notes and complete all practice exams
-Create a study guide with key points from each subject
-Review correct answers to practice exams
-Take a practice exam under timed conditions
-Identify any areas that need further study
-Complete a final review of all lecture notes

1 day before exams:

-Complete a final review of all lecture notes
-Identify any areas that need further study
-Take a practice exam under timed conditions
-Relax and get a good night's sleep
```

What did I learn?





- How to craft prompts that effectively communicate what I want generated
  - Providing context, specifications, and keywords
  - Iteratively refining the prompt to get the best possible result
- The wide variety of Chat GPT’s uses
  - ie. Generating code, creating a schedule, etc.

Thank you!