1Q. Create a simple Chatbot using nltk in python.

```
# Install NLTK
!pip install nltk

# Import necessary libraries
import nltk
from nltk.chat.util import Chat, reflections

# Download necessary NLTK data
nltk.download('punkt')

# Define some patterns and responses
patterns = [
    (r'hi|hello|hey', ['Hello!', 'Hi there!', 'Hey!']),
    (r'how are you?', ['I\'m good, thank you!', 'I\'m doing well, thanks
for asking.']),
    (r'what is your name?', ['You can call me ChatBot.', 'I go by the name
ChatBot.']),
    (r'(.*) your name?', ['You can call me ChatBot.', 'I go by the name
ChatBot.']),
    (r'bye|exit', ['Goodbye!', 'See you later!']),
]

# Create a chatbot
chatbot = Chat(patterns, reflections)

# Start the conversation
print("Hello! I'm ChatBot. How can I help you today? Type 'exit' to end
the chat.")
while True:
    user_input = input("You: ")
    if user_input.lower() in ['exit', 'bye']:
        print("ChatBot: Goodbye!")
        break
    response = chatbot.respond(user_input)
    print("ChatBot:", response)
```

output:

```
You: hi
ChatBot: Hi there!
You: how r u?
ChatBot: None
You: what is your name
ChatBot: You can call me ChatBot.
You: bye
ChatBot: Goodbye!
```

2Q. How would you create a prompt for summarizing a long article?

Understanding the Purpose of Summarization

Before crafting the prompt, it's crucial to understand the goal of summarization. The aim is to condense the original content into a shorter form that retains the main ideas and key details. This involves distinguishing between essential information and extraneous details.

Key Components of an Effective Summarization Prompt

Clarity: The prompt should clearly state what is expected from the summarization process.

Context: Provide context about the article, such as its main themes or topics, to guide the summarization.

Format: Specify how you want the summary to be structured (e.g., paragraph form, bullet points).

Length: Indicate the desired length of the summary (e.g., one paragraph, two sentences).

Tone and Style: Mention any specific tone or style preferences (e.g., formal, informal, technical).

Example Prompt Structure

Here's a structured example of how to create a prompt for summarizing a long article:

Step 1: Introduce the Task

Start with a clear instruction about what you want the model to do.

text

Please summarize the following article.

Step 2: Provide Context

Include a brief description of the article's main topic or purpose to give context.

text

The article discusses recent advancements in renewable energy technologies, focusing on solar and wind energy innovations.

Step 3: Specify Requirements

Clearly outline what you want in the summary.

text

Your summary should include:

- The main advancements discussed.

- Key statistics or figures mentioned.

- Implications for future energy policies.

Step 4: Define Format and Length

Specify how you would like the summary to be formatted and its length.

text

Please provide your summary in paragraph form, approximately 150-200 words long.

Step 5: Example of a Complete Prompt

Combining all elements, here’s what a complete prompt might look like:

text

Please summarize the following article:

[Insert Article Text Here]

The article discusses recent advancements in renewable energy technologies, focusing on solar and wind energy innovations. Your summary should include:

- The main advancements discussed.

- Key statistics or figures mentioned.

- Implications for future energy policies.

Please provide your summary in paragraph form, approximately 150-200 words long.

3Q. write a python code to text to text generation by giving prompt.

```
# Install the transformers library
!pip install transformers

# Import the necessary library
from transformers import pipeline

# Load the text generation model
text_generator = pipeline("text-generation", model="gpt2")

# Define the input prompt
prompt = " a scientist discovers,"

# Generate text based on the prompt
generated_text = text_generator(prompt, max_length=100,
num_return_sequences=1)

# Print the generated text
print(generated_text[0]['generated_text'])
```

OUTPUT:

```
a scientist discovers, the reason why water freezes is not the "real"
reason, but simply by means of the natural process called "water
retention." It would take more than just warming and cooling to hold water
in the same temperature range. That is, at least until the water that was
once "cold" breaks free of the ice sheet melts.
```

This new study, published in Nature Geoscience, is the first published analysis of the long-standing ice-sheet dynamics in ancient marine

4Q. write a python code to text to image generation by giving prompt with Image Saving

```
# Install the required libraries
!pip install diffusers transformers torch

# Import necessary libraries
import torch
from diffusers import StableDiffusionPipeline
from PIL import Image
from IPython.display import display

# Load the Stable Diffusion model (ensure you have a Hugging Face token)
model_id = "CompVis/stable-diffusion-v1-4"
pipe = StableDiffusionPipeline.from_pretrained(model_id,
torch_dtype=torch.float16).to("cuda")

# Define the input text prompt
prompt = "parrot on tree of size 100 by 100 pixel"

# Generate the image based on the prompt
with torch.no_grad():
    image = pipe(prompt).images[0]

# Display the generated image in the notebook
display(image)

# Save the generated image to a file
image.save("generated_image.png")  # You can change the filename and
format as needed

print("Image saved as 'generated_image.png'")
```

OUTPUT:

100% 50/50 [00:07<0

Image saved as 'generated_image.png'

FOR

```
prompt = "parrot on tree”
```