1. **Implement a basic N-gram model for text generation. For example, generate text using a bigram model using python.**

**Aim:**

To implement a basic N-Gram model for text generation.

**Code:**

import random

from collections import defaultdict

def build\_bigram\_model(text):

words = text.split()

bigrams = list(zip(words, words[1:]))

model = defaultdict(list)

for w1, w2 in bigrams:

model[w1].append(w2)

return model

def generate\_text(model, start\_word, length=10):

current\_word = start\_word

generated\_text = [current\_word]

for \_ in range(length - 1):

if current\_word in model:

next\_word = random.choice(model[current\_word])

generated\_text.append(next\_word)

current\_word = next\_word

else:

break

return " ".join(generated\_text)

user\_input = input("Enter a text sample: ")

bigram\_model = build\_bigram\_model(user\_input)

start\_word = random.choice(user\_input.split())

generated\_text = generate\_text(bigram\_model, start\_word)

print("\nGenerated Text:")

print(generated\_text)

**Input:**

Enter a text sample: the quick brown fox jumps over the lazy dog and the dog runs away

**Output:**

Generated Text:

jumps over the dog and the quick brown fox jumps

