39. Write a python program that can perform a letter frequency attack on an additive cipher without human intervention. Your software should produce possible plaintexts in rough order of likelihood. It would be good if your user interface allowed the user to specify “give me the top 10 possible plaintexts.”

Code:

```
import string
from collections import Counter
# English letter frequency order (most common first)
english_freq_order = "ETAOINSHRDLCUMWFGYPBVKJXQZ"
# Score plaintext by letter frequency match
def score_text(text):
    text = text.upper()
    letters_only = [c for c in text if c in string.ascii_uppercase]
    freq = Counter(letters_only)
    score = 0
    for i, ch in enumerate(english_freq_order):
        if ch in freq:
            # weight more common letters higher
            score += freq[ch] * (26 - i)
    return score
# Decrypt additive cipher with given shift
def decrypt_caesar(ciphertext, shift):
    plaintext = ""
    for ch in ciphertext:
        if ch.upper() in string.ascii_uppercase:
            val = (ord(ch.upper()) - ord('A') - shift) % 26
            if ch.isupper():
                plaintext += chr(val + ord('A'))
            else:
```

```
        plaintext += chr(val + ord('a'))
        else:
        plaintext += ch
    return plaintext
# Frequency attack
def freq_attack_additive(ciphertext, top_n=10):
    candidates = []
    for shift in range(26):
        plaintext = decrypt_caesar(ciphertext, shift)
        s = score_text(plaintext)
        candidates.append((s, shift, plaintext))
    # Sort by score descending
    candidates.sort(reverse=True)
    return candidates[:top_n]
# ------------------ Main ------------------
ciphertext = input("Enter additive cipher text:\n")
n = int(input("How many top plaintext guesses do you want? "))
results = freq_attack_additive(ciphertext, n)
print("\nTop", n, "likely plaintexts:\n")
for i, (score, shift, plaintext) in enumerate(results, 1):
    print(f"{i}. Shift {shift}: {plaintext}")
```

```
>>>
============== RESTART: C:/Users/Maria/OneDrive/Documents/ex39.py ==============
Enter additive cipher text:
Security
How many top plaintext guesses do you want? 2

Top 2 likely plaintexts:

1. Shift 0: Security
2. Shift 16: Comebsdi
>>>
```