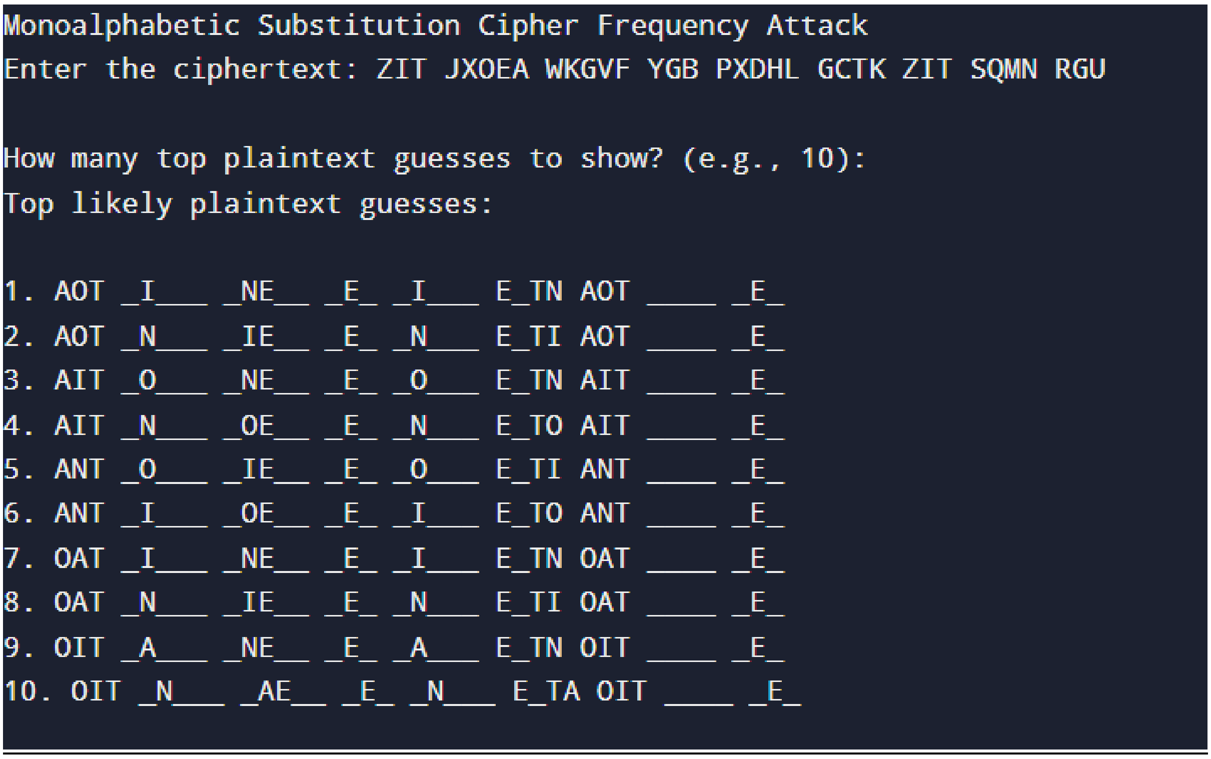
25) Write a python program that can perform a letter frequency attack on any monoalphabetic substitution cipher without human intervention. Your software should produce possible plaintexts in rough order of likelihood. It would be good if your user interface allowed the user to specify “give me the top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

PROGRAM:-

```
import string

from collections import Counter

from itertools import permutations

ENGLISH_FREQ_ORDER = "ETAOINSHRDLCUMWFGYPBVKJXQZ"

def clean_text(text):

    """Removes non-alpha characters and converts to uppercase."""

    return ''.join([c for c in text.upper() if c.isalpha()])

def frequency_attack(ciphertext, top_n=10):

    cleaned = clean_text(ciphertext)

    cipher_freq = Counter(cleaned)

    cipher_letters_sorted = ''.join([pair[0] for pair in cipher_freq.most_common()])

    trial_letters = 6

    cipher_top = cipher_letters_sorted[:trial_letters]

    english_top = ENGLISH_FREQ_ORDER[:trial_letters]

    guesses = []

    seen = set()

    for perm in permutations(english_top):

        # Build substitution map

        sub_map = {c: p for c, p in zip(cipher_top, perm)}

        for c in string.ascii_uppercase:

            if c not in sub_map:

                sub_map[c] = '_'

        decrypted = ""

        for ch in ciphertext.upper():

            if ch in string.ascii_uppercase:

                decrypted += sub_map[ch]

            else:
                decrypted += ch
        if decrypted not in seen:
            seen.add(decrypted)
            guesses.append(decrypted)
        if len(guesses) >= top_n * 2:  # Generate a few more than needed
            break
    return guesses[:top_n]
def main():
    print("Monoalphabetic Substitution Cipher Frequency Attack")
    ciphertext = input("Enter the ciphertext: ")
    try:
        top_n = int(input("How many top plaintext guesses to show? (e.g., 10): "))
    except ValueError:
        top_n = 10
    print("\nTop likely plaintext guesses:\n")
    guesses = frequency_attack(ciphertext, top_n)
    for i, guess in enumerate(guesses, 1):
        print(f"{i}. {guess}")
if __name__ == "__main__":
    main()
```

OUTPUT:-



```
Monoalphabetic Substitution Cipher Frequency Attack
Enter the ciphertext: ZIT JXOEA WKGVF YGB PXDHL GCTK ZIT SQMN RGU

How many top plaintext guesses to show? (e.g., 10):
Top likely plaintext guesses:

1. AOT _I___ _NE__ _E_ _I___ E_TN AOT ____ _E_
2. AOT _N___ _IE__ _E_ _N___ E_TI AOT ____ _E_
3. AIT _O___ _NE__ _E_ _O___ E_TN AIT ____ _E_
4. AIT _N___ _OE__ _E_ _N___ E_TO AIT ____ _E_
5. ANT _O___ _IE__ _E_ _O___ E_TI ANT ____ _E_
6. ANT _I___ _OE__ _E_ _I___ E_TO ANT ____ _E_
7. OAT _I___ _NE__ _E_ _I___ E_TN OAT ____ _E_
8. OAT _N___ _IE__ _E_ _N___ E_TI OAT ____ _E_
9. OIT _A___ _NE__ _E_ _A___ E_TN OIT ____ _E_
10. OIT _N___ _AE__ _E_ _N___ E_TA OIT ____ _E_
```