24) 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):
    return ''.join(c for c in text.upper() if c.isalpha())
def build_substitution_map(cipher_letters, english_letters):
    return dict(zip(cipher_letters, english_letters))
def apply_substitution(ciphertext, substitution_map):
    result = ""
    for c in ciphertext.upper():
        if c in substitution_map:
            result += substitution_map[c]
        elif c in string.ascii_uppercase:
            result += '_'
        else:
            result += c  # preserve spaces/punctuation
    return result
def frequency_attack(ciphertext, top_n=5):
    cleaned = clean_text(ciphertext)
    cipher_freq = Counter(cleaned)
    most_common = [pair[0] for pair in cipher_freq.most_common()]
    limit = min(6, len(most_common))
    cipher_top = most_common[:limit]
    english_top = ENGLISH_FREQ_ORDER[:limit]
    guesses = []
    for perm in permutations(english_top):
```

```
        sub_map = build_substitution_map(cipher_top, perm)
        guess = apply_substitution(ciphertext, sub_map)
        guesses.append(guess)
    unique = list(dict.fromkeys(guesses))
    return unique[:top_n]
def main():
    print("Monoalphabetic Substitution Cipher Frequency Attack")
    try:
        ciphertext = input("Enter ciphertext: ").strip()
        if not ciphertext:
            raise ValueError("Empty input")
        top_n = int(input("How many top plaintext guesses to show? (e.g., 10): ").strip())
    except Exception:
        print("Invalid input. Using default values.")
        ciphertext = "ZIT JXOEA WKGVF YGB PXDHL GCTK ZIT SQMN RGU"
        top_n = 5
    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 ciphertext: ZIT JXOEA WKGVF YGB PXDHL GCTK ZIT SQMN RGU
How many top plaintext guesses to show? (e.g., 10): 5

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_

=== Code Execution Successful ===
```