16. 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.”

```
# Simple monoalphabetic substitution cipher frequency attack
english = "ETAOINSHRDLCUMWFGYPBVKJXQ"
```

Code:

```
def score(t):
    t = t.upper()
    return sum(t.count(c) * (26 - i) for i, c in enumerate(english))
cipher = input("Enter ciphertext: ")
top_n = int(input("Top N plaintexts? "))
# Count ciphertext letter frequency
freq = {}
for c in cipher.upper():
    if c.isalpha():
        freq[c] = freq.get(c, 0) + 1
# Cipher letters sorted by frequency
cipher_order = "".join(sorted(freq, key=freq.get, reverse=True))
# Build simple substitution map
mapping = {cipher_order[i]: english[i] for i in range(len(cipher_order))}
# Create plaintext
plain = ""
for ch in cipher:
    if ch.upper() in mapping:
        p = mapping[ch.upper()]
        plain += p.lower() if ch.islower() else p
    else:
```

```
        plain += ch

print("\nMost likely plaintext:\n")

print(plain)
```