15. 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.”

```
# Letter-frequency attack for additive (Caesar) cipher
# English letter frequency order (most to least common)
freq_order = "ETAOINSHRDLCUMWFGYPBVKJXQ"
```

Code:

```
def score(text):
    text = text.upper()
    count = {c: text.count(c) for c in freq_order}
    # Score = sum of rank weights for frequent letters
    return sum((26-i) * count[c] for i, c in enumerate(freq_order))
def decrypt(cipher, shift):
    res = ""
    for ch in cipher:
        if ch.isalpha():
            base = ord('A') if ch.isupper() else ord('a')
            res += chr((ord(ch) - base - shift) % 26 + base)
        else:
           res += ch
    return res
ciphertext = input("Enter ciphertext: ")
top_n = int(input("How many top plaintexts? "))
candidates = []
for s in range(26):
    p = decrypt(ciphertext, s)
    candidates.append((score(p), s, p))
```

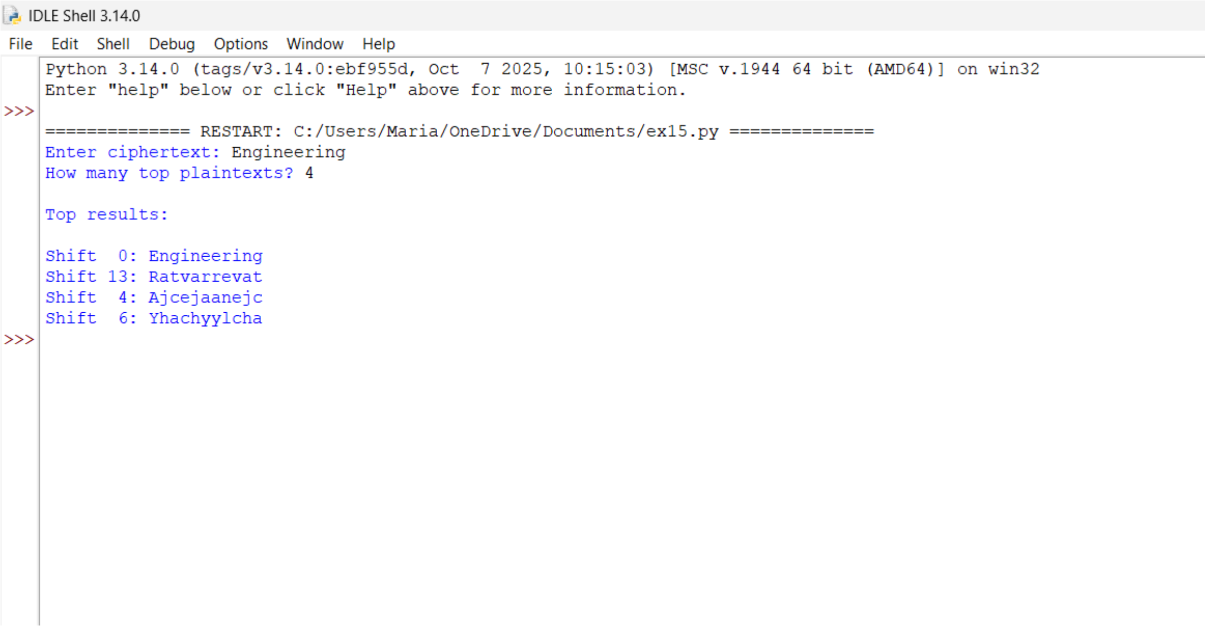
```
# Sort by score (best first)

candidates.sort(reverse=True)

print("\nTop results:\n")

for i in range(top_n):

    print(f"Shift {candidates[i][1]:2d}: {candidates[i][2]}")
```



```
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct  7 2025, 10:15:03) [MSC v.1944 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>>
============== RESTART: C:/Users/Maria/OneDrive/Documents/ex15.py ==============
Enter ciphertext: Engineering
How many top plaintexts? 4

Top results:

Shift  0: Engineering
Shift 13: Ratvarrevat
Shift  4: Ajcejaanejc
Shift  6: Yhachyylcha
>>>
```