main.c

Run

```c
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <stdlib.h>

#define MAX_LEN 1000
#define NUM_SHIFTS 26

// English letter frequency (relative,
    normalized for scoring)
float english_freq[26] = {
    8.167, 1.492, 2.782, 4.253, 12.702, 2.228,
    2.015, 6.094, 6.966, 0.153, 0.772, 4.025,
    2.406, 6.749, 7.507, 1.929, 0.095, 5.987,
    6.327, 9.056, 2.758, 0.978, 2.360, 0.150,
    1.974, 0.074
};

```

Output

Clear

```
Enter the ciphertext (uppercase letters only):
WKH HDJOH KDV ODQGHG
Enter number of top plaintexts to show: 5

Top 5 possible plaintexts:
Shift:  3 | Score: -565.42 | Plaintext: THE EAGLE HAS
    LANDED
Shift: 25 | Score: -905.00 | Plaintext: XLI IEKPI LEW
    PERHIH
Shift: 14 | Score: -984.96 | Plaintext: IWT TPVAT WPH
    APCSTS
Shift:  2 | Score: -1097.76 | Plaintext: UIF FBHMF IB
    MBOEFE
Shift: 10 | Score: -1100.17 | Plaintext: MAX XTZEX AT
    ETGWXW


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

main.c

Run

```
28
29      float score = 0.0;
30      if (total == 0) return 0.0;
31
32      for (int i = 0; i < 26; i++) {
33          float observed = (float)count[i] / total
                * 100;
34          float expected = english_freq[i];
35          float diff = observed - expected;
36          score -= diff * diff;  // Squared error
37      }
38      return score;
39  }
40
41  // Caesar decrypt with a shift
42  void caesarDecrypt(char *ciphertext, int key,
        char *plaintext) {
```

Output

Clear

```
Enter the ciphertext (uppercase letters only):
WKH HDJOH KDV ODQGHG
Enter number of top plaintexts to show: 5

Top 5 possible plaintexts:
Shift:  3 | Score: -565.42 | Plaintext: THE EAGLE HAS
    LANDED
Shift: 25 | Score: -905.00 | Plaintext: XLI IEKPI LEW
    PERHIH
Shift: 14 | Score: -984.96 | Plaintext: IWT TPVAT WPH
    APCSTS
Shift:  2 | Score: -1097.76 | Plaintext: UIF FBHMF IBT
    MBOEFE
Shift: 10 | Score: -1100.17 | Plaintext: MAX XTZEX ATL
    ETGWXW


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

main.c

Run

```
}

// Sort candidates by score
qsort(candidates, NUM_SHIFTS, sizeof
    (Candidate), compare);

printf("\nTop %d possible plaintexts:\n",
    topN);
for (int i = 0; i < topN && i < NUM_SHIFTS;
    i++) {
    printf("Shift: %2d | Score: %6.2f |
        Plaintext: %s\n",
            candidates[i].shift,
                candidates[i].score,
                candidates[i].plaintext);
}

return 0;
```

Output

Clear

```
Enter the ciphertext (uppercase letters only):
WKH HDJOH KDV ODQGHG
Enter number of top plaintexts to show: 5

Top 5 possible plaintexts:
Shift:  3 | Score: -565.42 | Plaintext: THE EAGLE HAS
    LANDED
Shift: 25 | Score: -905.00 | Plaintext: XLI IEKPI LEW
    PERHIH
Shift: 14 | Score: -984.96 | Plaintext: IWT TPVAT WPH
    APCSTS
Shift:  2 | Score: -1097.76 | Plaintext: UIF FBHMF IBT
    MBOEFE
Shift: 10 | Score: -1100.17 | Plaintext: MAX XTZEX ATL
    ETGWXW


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