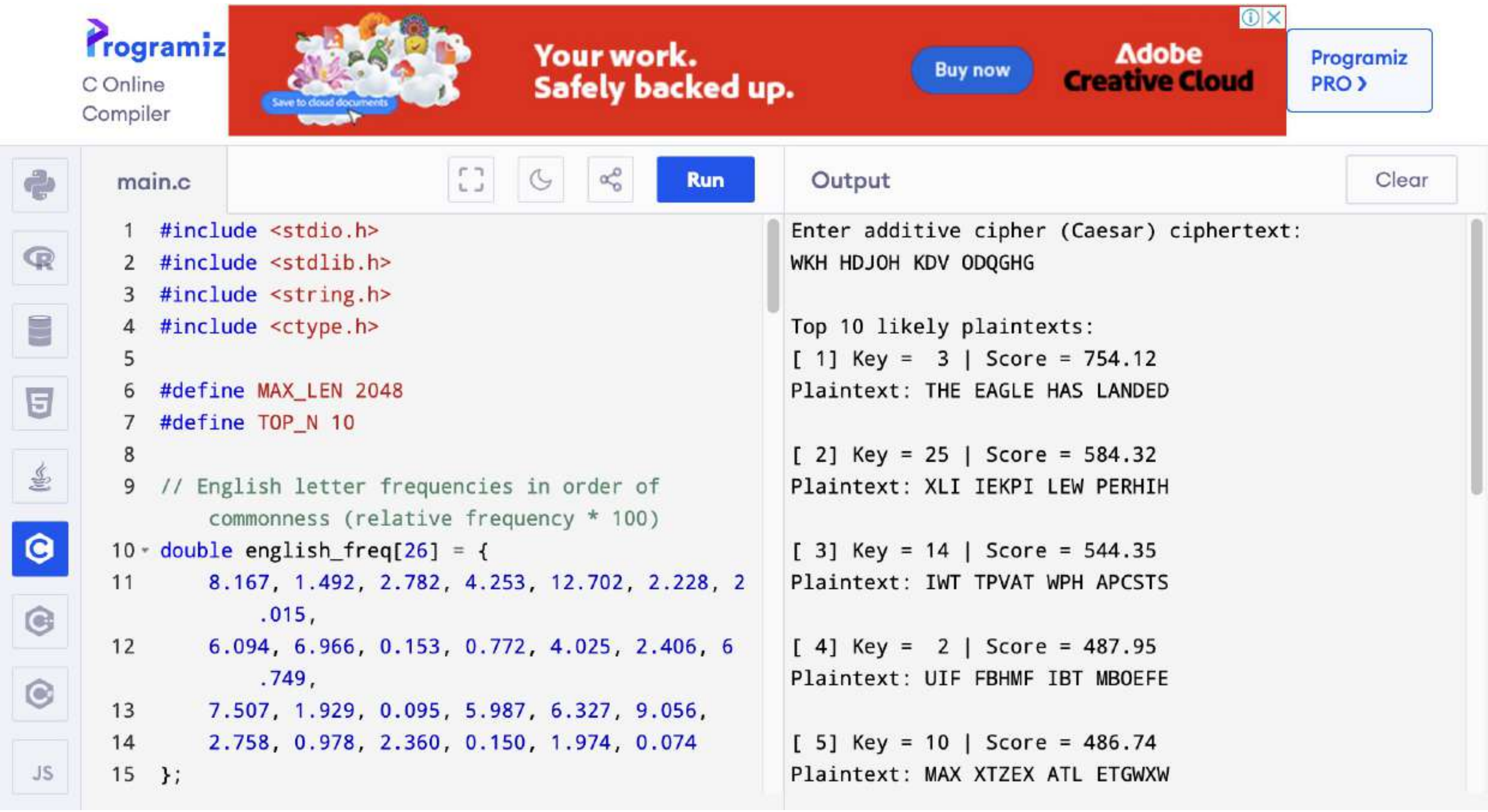
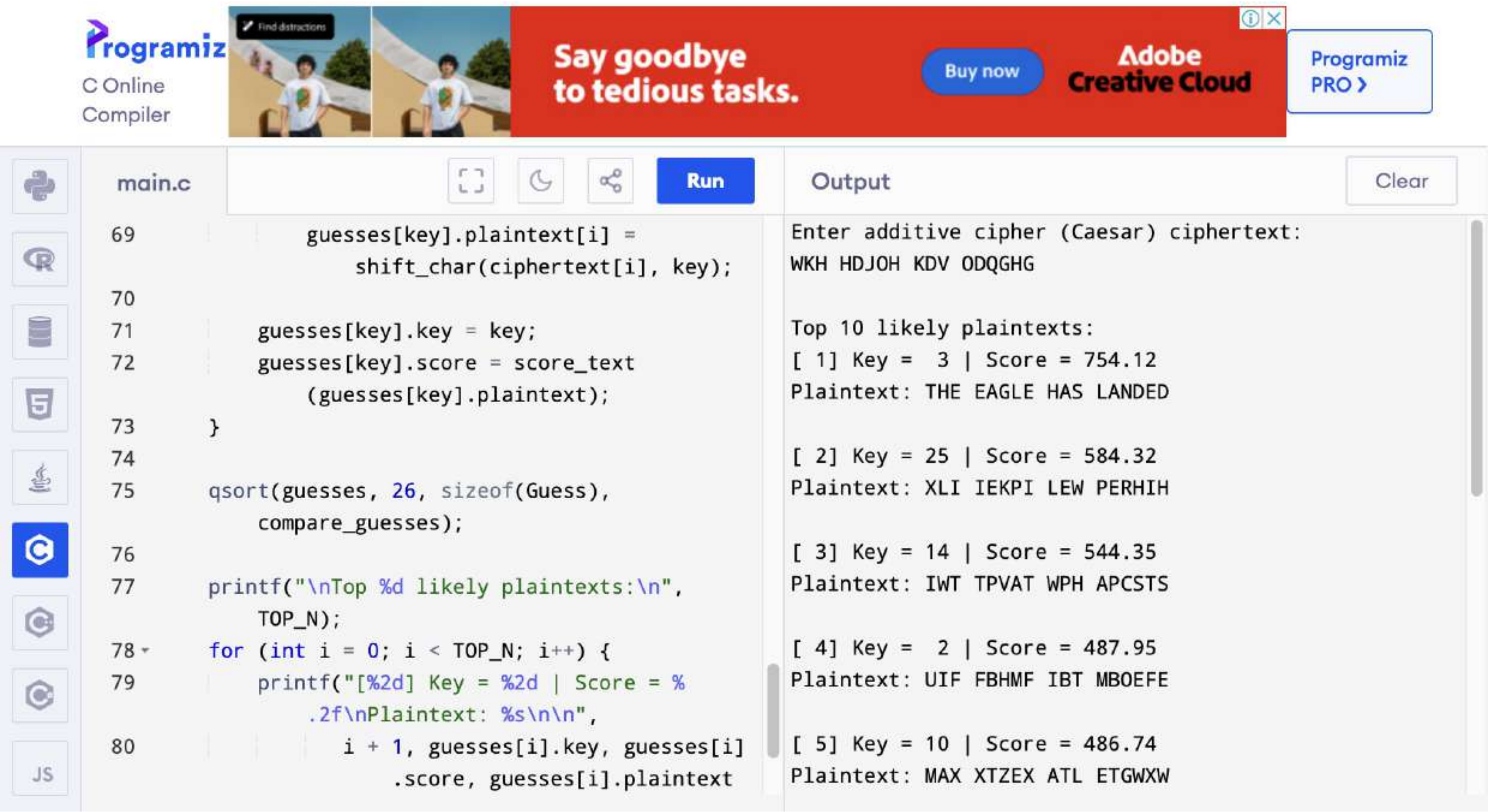


```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

#define MAX_LEN 2048
#define TOP_N 10

// English letter frequencies in order of commonness (relative frequency * 100)
double 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
};
```

```
Enter additive cipher (Caesar) ciphertext:
WKH HDJOH KDV ODQGHG

Top 10 likely plaintexts:
[ 1] Key =  3 | Score = 754.12
Plaintext: THE EAGLE HAS LANDED

[ 2] Key = 25 | Score = 584.32
Plaintext: XLI IEKPI LEW PERHIH

[ 3] Key = 14 | Score = 544.35
Plaintext: IWT TPVAT WPH APCSTS

[ 4] Key =  2 | Score = 487.95
Plaintext: UIF FBHMF IBT MBOEFE

[ 5] Key = 10 | Score = 486.74
Plaintext: MAX XTZEX ATL ETGWXW
```



```c
            guesses[key].plaintext[i] = shift_char(ciphertext[i], key);

        guesses[key].key = key;
        guesses[key].score = score_text(guesses[key].plaintext);
    }

    qsort(guesses, 26, sizeof(Guess), compare_guesses);

    printf("\nTop %d likely plaintexts:\n", TOP_N);
    for (int i = 0; i < TOP_N; i++) {
        printf("[%2d] Key = %2d | Score = %.2f\nPlaintext: %s\n\n",
               i + 1, guesses[i].key, guesses[i].score, guesses[i].plaintext
```

```
Enter additive cipher (Caesar) ciphertext:
WKH HDJOH KDV ODQGHG

Top 10 likely plaintexts:
[ 1] Key =  3 | Score = 754.12
Plaintext: THE EAGLE HAS LANDED

[ 2] Key = 25 | Score = 584.32
Plaintext: XLI IEKPI LEW PERHIH

[ 3] Key = 14 | Score = 544.35
Plaintext: IWT TPVAT WPH APCSTS

[ 4] Key =  2 | Score = 487.95
Plaintext: UIF FBHMF IBT MBOEFE

[ 5] Key = 10 | Score = 486.74
Plaintext: MAX XTZEX ATL ETGWXW
```