



main.c

cipher text

```
41
42 // Frequency analysis: "B" is most frequent,
    "U" is second most frequent
43 // Let's assume that "B" corresponds to 'E'
    and "U" corresponds to 'T'
44 // Therefore, we need to solve for 'a' and
    'b' such that:
45 //  $E = (a * B + b) \% 26$ 
46 //  $T = (a * U + b) \% 26$ 
47
48 // Mapping from "B" -> 'E' and "U" -> 'T'
49 int b_freq = 'B' - 'A'; // Position of B in
    alphabet (B = 1)
50 int e_freq = 'E' - 'A'; // Position of E in
    alphabet (E = 4)
51 int u_freq = 'U' - 'A'; // Position of U in
    alphabet (U = 20)
52 int t_freq = 'T' - 'A'; // Position of T in
```

Run

Output

Clear

A module you have imported isn't available at the moment  
. It will be available soon.



main.c

Run

Output

Clear

```
53
54 // Solving the system of equations:
55 // E = (a * B + b) % 26
56 // T = (a * U + b) % 26
57 // We have two equations and two unknowns (a
    and b).
58
59 // Equation 1: E = (a * B + b) % 26 -> 4 =
    (a * 1 + b) % 26
60 // Equation 2: T = (a * U + b) % 26 -> 19 =
    (a * 20 + b) % 26
61
62 // Step 1: Solve for b in terms of a using
    the first equation
63 // b = (E - a * B) % 26
64 int a, b;
65 for (a = 1; a < 26; a++) {
66     if (a % 2 != 0) { // Ensure 'a' is
```

A module you have imported isn't available at the moment  
. It will be available soon.



main.c



Run

Output

Clear

```
        {
69         break; // Solve for a and b
70     }
71 }
72 }
73
74 if (a < 26) {
75     printf("Found keys: a = %d, b = %d\n", a
76         , b);
77     affine_decrypt(ciphertext, a, b);
78 } else {
79     printf("No valid solution found for 'a'
80         and 'b'.\n");
81 }
82 return 0;
83 }
```

A module you have imported isn't available at the moment  
. It will be available soon.