



main.c



Run

Output

Clear

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <ctype.h>
4 int modInverse(int a, int m) {
5     a = a % m;
6     for (int x = 1; x < m; x++) {
7         if ((a * x) % m == 1)
8             return x;
9     }
10    return -1;
11 }
12 void affineEncrypt(char *plaintext, char
    *ciphertext, int a, int b) {
13     for (int i = 0; plaintext[i] != '\0'; i++) {
14         char ch = plaintext[i];
15         if (isalpha(ch)) {
16             int p = toupper(ch) - 'A';
17             int c = (a * p + b) % 26;
```

```
Enter value of 'a' (must be coprime to 26): 5
Enter value of 'b' (0 - 25): 8
Enter plaintext: affine cipher
Encrypted text: ihhwvc swfrcp
Decrypted text: affine cipher
```

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



main.c



Run

Output

Clear

```
19-     } else {
20-         ciphertext[i] = ch;
21-     }
22- }
23- ciphertext[strlen(plaintext)] = '\0';
24- }
25- void affineDecrypt(char *ciphertext, char
    *plaintext, int a, int b) {
26-     int a_inv = modInverse(a, 26);
27-     if (a_inv == -1) {
28-         printf("Invalid key 'a'. No modular
            inverse exists.\n");
29-         return;
30-     }
31-
32-     for (int i = 0; ciphertext[i] != '\0'; i++)
33-     {
34-         char ch = ciphertext[i];
```

Enter value of 'a' (must be coprime to 26): 5  
Enter value of 'b' (0 - 25): 8  
Enter plaintext: affine cipher  
Encrypted text: ihhwvc swfrcp  
Decrypted text: affine cipher

=== Code Execution Successful ===



main.c

Run

```
        to 26.\n");
60     return 1;
61 }
62
63 printf("Enter value of 'b' (0 - 25): ");
64 scanf("%d", &b);
65 getchar();
66 printf("Enter plaintext: ");
67 fgets(plaintext, sizeof(plaintext), stdin);
68 plaintext[strcspn(plaintext, "\n")] = '\0';
69 affineEncrypt(plaintext, ciphertext, a, b);
70 printf("Encrypted text: %s\n", ciphertext);
71 affineDecrypt(ciphertext, decrypted, a, b);
72 printf("Decrypted text: %s\n", decrypted);
73
74 return 0;
75 }
76
```

Output

Clear

Enter value of 'a' (must be coprime to 26): 5  
Enter value of 'b' (0 - 25): 8  
Enter plaintext: affine cipher  
Encrypted text: ihhwvc swfrcp  
Decrypted text: affine cipher  
  
=== Code Execution Successful ===