

main.c

Run

Output

Clear

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <ctype.h>
4 #define ALPHABET_SIZE 26
5 void generate_cipher_alphabet(const char
    *keyword, char *cipher_alphabet) {
6     int used[ALPHABET_SIZE] = {0};
7     int index = 0;
8     for (int i = 0; keyword[i] != '\0'; i++) {
9         char ch = toupper(keyword[i]);
10        if (ch >= 'A' && ch <= 'Z' && !used[ch -
            'A']) {
11            cipher_alphabet[index++] = ch;
12            used[ch - 'A'] = 1;
13        }
14    }
15    for (char ch = 'A'; ch <= 'Z'; ch++) {
16        if (!used[ch - 'A']) {
```

Cipher Alphabet: C I P H E R A B D F G J K L M N O Q S T
U V W X Y Z
Plaintext: HELLO WORLD
Ciphertext: BEJJM WMQJH

=== Code Execution Successful ===

main.c

```
19 }
20 }
21 void encrypt(const char *plaintext, const char
    *cipher_alphabet, char *ciphertext) {
22     for (int i = 0; plaintext[i] != '\0'; i++) {
23         char ch = toupper(plaintext[i]);
24
25         if (ch >= 'A' && ch <= 'Z') {
26             ciphertext[i] = cipher_alphabet[ch -
                'A'];
27         } else {
28             ciphertext[i] = plaintext[i];
29         }
30     }
31 }
32 void print_cipher_alphabet(const char
    *cipher_alphabet) {
33     printf("Cipher Alphabet: ");
34     for (int i = 0; i < ALPHABET_SIZE; i++) {
```

Output

Clear

```
Cipher Alphabet: C I P H E R A B D F G J K L M N O Q S T
                U V W X Y Z
Plaintext: HELLO WORLD
Ciphertext: BEJJM WMQJH
```

=== Code Execution Successful ===

main.c

Run

Output

Clear

```
36     }
37     printf("\n");
38 }
39 int main() {
40     const char *keyword = "CIPHER";
41     char cipher_alphabet[ALPHABET_SIZE];
42     generate_cipher_alphabet(keyword,
43                             cipher_alphabet);
44     print_cipher_alphabet(cipher_alphabet);
45     const char *plaintext = "HELLO WORLD";
46     char ciphertext[strlen(plaintext) + 1];
47     encrypt(plaintext, cipher_alphabet,
48            ciphertext);
49     printf("Plaintext: %s\n", plaintext);
50     printf("Ciphertext: %s\n", ciphertext);
51     return 0;
52 }
```

Cipher Alphabet: C I P H E R A B D F G J K L M N O Q S T
U V W X Y Z
Plaintext: HELLO WORLD
Ciphertext: BEJJM WMQJH

=== Code Execution Successful ===