

Program 1: Plain text to Cypher Text

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
void main()
{char plain[10], cipher[10];
int key , i ,length;
int result;
printf("\n Enter the plain text:");
scanf("%s",plain);
printf("\n Enter the key value:");
scanf("%d",&key);
printf(" \n \n \t PLAIN TEXT: %s",plain);
printf("\n \n \t ENCRYPTED TEXT :");
for(i=0, length=strlen(plain);i<length;i++)
{
cipher[i] = plain[i] + key;
if (isupper(plain[i]&&(cipher[i]>'Z'))
cipher[i] = cipher[i] - 26;
if(slower(plain[i])&&(cipher[i]>'z'))
cipher[i] = cipher[i] -26;
printf("%c",cipher[i]);
}
printf("\n \n \t AFTER DECRYPTION:");
for(i=0,i<length;i++){
plain[i] =cipher[i] -key;
if(slower(cipher[i])&&(plain[i]<'A'))
plain[i]=plain[i] -26;
if(slower(cipher[i])&&(plain[i]<'a'))
plain[i]=plain[i] -26;
printf("%c",plain[i]);
}
}
```

Program 2:Plain text to Cypher Text

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

void main() {
    char plain[10], cipher[10];
    int key, i, length;

    printf("\nEnter the plain text: ");
    scanf("%s", plain);

    printf("\nEnter the key value: ");
    scanf("%d", &key);
```

```

printf("\n\n\tPLAIN TEXT: %s", plain);
printf("\n\n\tENCRYPTED TEXT: ");

length = strlen(plain);

// Encryption process
for (i = 0; i < length; i++) {
    if (isalpha(plain[i])) {
        cipher[i] = plain[i] + key;

        // Handle uppercase letters
        if (isupper(plain[i]) && cipher[i] > 'Z') {
            cipher[i] = cipher[i] - 26;
        }
        // Handle lowercase letters
        if (islower(plain[i]) && cipher[i] > 'z') {
            cipher[i] = cipher[i] - 26;
        }
    } else {
        cipher[i] = plain[i]; // Non-alphabetic characters remain unchanged
    }

    printf("%c", cipher[i]);
}
cipher[length] = '\0'; // Null-terminate the cipher text

printf("\n\n\tAFTER DECRYPTION: ");

// Decryption process
for (i = 0; i < length; i++) {
    if (isalpha(cipher[i])) {
        plain[i] = cipher[i] - key;

        // Handle uppercase letters
        if (isupper(cipher[i]) && plain[i] < 'A') {
            plain[i] = plain[i] + 26;
        }
        // Handle lowercase letters
        if (islower(cipher[i]) && plain[i] < 'a') {
            plain[i] = plain[i] + 26;
        }
    } else {
        plain[i] = cipher[i]; // Non-alphabetic characters remain unchanged
    }

    printf("%c", plain[i]);
}
plain[length] = '\0'; // Null-terminate the plain text
}

```

```
exp1.c x *expp1.c x
1 #include <stdio.h>
2 #include <string.h>
3 #include <ctype.h>
4 void main() {
5     char plain[10], cipher[10];
6     int key, i, length;
7     printf("\nEnter the plain text: ");
8     scanf("%s", plain);
9     printf("\nEnter the key value: ");
10    scanf("%d", &key);
11    printf("\n\n\tPLAIN TEXT: %s", plain);
12    printf("\n\n\tENCRYPTED TEXT: ");
13    length = strlen(plain);
14    // Encryption process
15    for (i = 0; i < length; i++) {
16        if (isalpha(plain[i])) {
17            cipher[i] = plain[i] + key;
18            // Handle uppercase letters
19            if (isupper(plain[i]) && cipher[i] > 'Z') {
20                cipher[i] = cipher[i] - 26;
21            }
22            // Handle lowercase letters
23            if (islower(plain[i]) && cipher[i] > 'z') {
24                cipher[i] = cipher[i] - 26;
25            }
26        } else {
27            cipher[i] = plain[i]; // Non-alphabetic characters remain unchanged
28        }
29    }
30    printf("\n\n\tCIPHER TEXT: ");
31    for (i = 0; i < length; i++) {
32        printf("%c", cipher[i]);
33    }
34    printf("\n\n\tAFTER DECRYPTION: ");
35    // Decryption process
36    for (i = 0; i < length; i++) {
37        if (isalpha(cipher[i])) {
38            plain[i] = cipher[i] - key;
39            // Handle uppercase letters
40            if (isupper(cipher[i]) && plain[i] < 'A') {
41                plain[i] = plain[i] + 26;
42            }
43            // Handle lowercase letters
44            if (islower(cipher[i]) && plain[i] < 'a') {
45                plain[i] = plain[i] + 26;
46            }
47        } else {
48            plain[i] = cipher[i]; // Non-alphabetic characters remain unchanged
49        }
50        printf("%c", plain[i]);
51    }
52    plain[length] = '\0'; // Null-terminate the plain text
53 }
```

```
exp1.c x *expp1.c x
26 } else {
27     cipher[i] = plain[i]; // Non-alphabetic characters remain unchanged
28 }
29
30 printf("%c", cipher[i]);
31 }
32 cipher[length] = '\0'; // Null-terminate the cipher text
33 printf("\n\n\tAFTER DECRYPTION: ");
34 // Decryption process
35 for (i = 0; i < length; i++) {
36     if (isalpha(cipher[i])) {
37         plain[i] = cipher[i] - key;
38
39         // Handle uppercase letters
40         if (isupper(cipher[i]) && plain[i] < 'A') {
41             plain[i] = plain[i] + 26;
42         }
43         // Handle lowercase letters
44         if (islower(cipher[i]) && plain[i] < 'a') {
45             plain[i] = plain[i] + 26;
46         }
47     } else {
48         plain[i] = cipher[i]; // Non-alphabetic characters remain unchanged
49     }
50     printf("%c", plain[i]);
51 }
52 plain[length] = '\0'; // Null-terminate the plain text
53 }
```

Output:

```
apsit@apsit-HP-280-G3-MT: ~/Desktop
File Edit View Search Terminal Help
apsit@apsit-HP-280-G3-MT:~$ cd Desktop
apsit@apsit-HP-280-G3-MT:~/Desktop$ gcc expp1.c
apsit@apsit-HP-280-G3-MT:~/Desktop$ ./a.out

Enter the plain text: hello

Enter the key value: 1

PLAIN TEXT: hello

ENCRYPTED TEXT: ifmmp

AFTER DECRYPTION: helloapsit@apsit-HP-280-G3-MT:~/Desktop$
```