

cipherText

Cipher text is a popular encryption technique. What we do in cipher text is that we can encrypt each alpha ('a' .. 'z', 'A' .. 'Z') character with +1. For example, "Hello" can be encrypted with +1 cipher to "Ifmmp". If a character is 'z' or 'Z', the corresponding encrypted character will be 'a' or 'A' respectively. For other characters, no encryption is performed. We use call by reference in the implementation. Write the C functions cipher() and decipher() with the following function prototypes:

```
void cipher(char *s);  
void decipher(char *s);
```

A sample program template is given below to test the functions:

```
#include <stdio.h>  
#include <string.h>  
#include <ctype.h>  
void cipher(char *s);  
void decipher(char *s);  
int main()  
{  
    char str[80], dummychar, *p;  
    int choice;  
  
    printf("Select one of the following options: \n");  
    printf("1: cipher() \n");  
    printf("2: decipher() \n");  
    printf("3: exit() \n");  
    do {  
        printf("Enter your choice: \n");  
        scanf("%d", &choice);  
        switch (choice) {  
            case 1:  
                scanf("%c",&dummychar);  
                printf("Enter the string: \n");  
                fgets(str, 80, stdin);  
                if (p=strchr(str,'\n')) *p = '\0';  
                printf("To cipher: %s -> ", str);  
                cipher(str);  
                printf("%s\n", str);  
                break;  
            case 2:  
                scanf("%c",&dummychar);  
                printf("Enter the string: \n");  
                fgets(str, 80, stdin);  
                if (p=strchr(str,'\n')) *p = '\0';  
                printf("To decipher: %s -> ", str);  
                decipher(str);  
                printf("%s\n", str);  
                break;  
            default:
```

```

        break;
    }
} while (choice < 3);
return 0;
}
void cipher(char *s)
{
    /* Write your code here */
}
void decipher(char *s)
{
    /* Write your code here */
}

```

Some sample input and output sessions are given below:

(1) Test Case 1:
Select one of the following options:

1: cipher()
2: decipher()
3: exit()

Enter your choice:

1

Enter the string:

123a

To cipher: 123a -> 123b

Enter your choice:

3

(2) Test Case 2:
Select one of the following options:

1: cipher()
2: decipher()
3: exit()

Enter your choice:

2

Enter the string:

123b

To decipher: 123b -> 123a

Enter your choice:

3

(3) Test Case 3:
Select one of the following options:

1: cipher()
2: decipher()
3: exit()

Enter your choice:

1

Enter the string:

abcxyz

To cipher: abcxyz -> bcdyza
Enter your choice:
2
Enter the string:
bcdyza
To decipher: bcdyza -> abcxyz
Enter your choice:
3

(4) Test Case 4:
Select one of the following options:
1: cipher()
2: decipher()
3: exit()
Enter your choice:
1
Enter the string:
HELLO Hello
To cipher: HELLO Hello -> IFMMP Ifmmp
Enter your choice:
2
Enter the string:
IFMMP Ifmmp
To decipher: IFMMP Ifmmp -> HELLO Hello
Enter your choice:
3

```
void cipher(char *s)
{
    int i;

    for(i=0;s[i]!='\0';i++)
    {
        if(s[i]>='a'&&s[i]<='y' || s[i]>='A'&&s[i]<='Y')
        {
            s[i] = s[i]+1;
            continue;
        }
        else if(s[i]=='z')
        {
            s[i] = 'a';
            continue;
        }
        else if(s[i]=='Z')
        {
            s[i] = 'A';
            continue;
        }
    }
}

void decipher(char *s)
{
    int i;

    for(i=0;s[i]!='\0';i++)
    {
        if(s[i]=='A')
        {
            s[i] = 'Z';
            continue;
        }
        else if(s[i]=='a')
        {
            s[i] = 'z';
            continue;
        }
        else if(s[i]>='b'&&s[i]<='z' || s[i]>='B'&&s[i]<='Z')
        {
            s[i] = s[i]-1;
            continue;
        }
    }
}
```