**7. Write a C program for monoalphabetic cipher is that both sender and receiver must commit the permuted cipher sequence to memory. A common technique for avoiding this is to use a keyword from which the cipher sequence can be generated.For example, using the keyword CIPHER, write out the keyword followed by unused letters in normal order and match this against the plaintext letters:**

**plain: a b c d e f g h i j k l m n o p q r s t u v w x y z**

**cipher: 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**

**CODE:**

**#include<stdio.h>**

**char monocipher\_encr(char);**

**char alpha[27][3] = { { 'a', 'c' }, { 'b', 'i' }, { 'c', 'p' }, { 'd', 'h' }, {**

**'e', 'e' }, { 'f', 'r' }, { 'g', 'a' }, { 'h', 'b' }, { 'i', 'd' }, {**

**'j', 'f' }, { 'k', 'g' }, { 'l', 'j' }, { 'm', 'k' }, { 'n', 'l' }, {**

**'o', 'm' }, { 'p', 'n' }, { 'q', 'o' }, { 'r', 'q' }, { 's', 's' }, {**

**'t', 't' }, { 'u', 'u' }, { 'v', 'v' }, { 'w', 'w' }, { 'x', 'x' }, {**

**'y', 'y' }, { 'z', 'z' } };**

**char str[20];**

**int main() {**

**char str[20], str2[20];**

**int i;**

**printf("\n enter a plaintext..");**

**gets(str);**

**for (i = 0; str[i]; i++) {**

**str2[i] = monocipher\_encr(str[i]);**

**}**

**str2[i] = '\0';**

**printf("\n Before Decryption..%s", str);**

**printf("\n After Decryption..%s\n", str2);**

**}**

**char monocipher\_encr(char a) {**

**int i;**

**for (i = 0; i < 27; i++) {**

**if (a == alpha[i][0])**

**break;**

**}**

**return alpha[i][1];**

**}**

**OUTPUT:**

