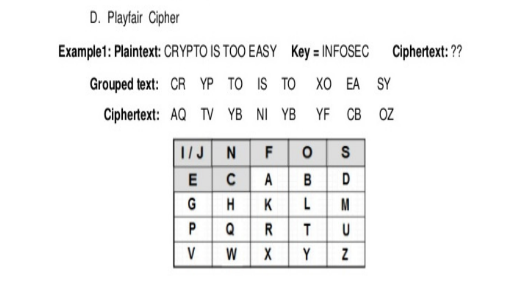
**AIM: To write a C program to implement the Playfair Substitution technique.**

**DESCRIPTION:** The Playfair cipher uses a 5x5 grid with unique letters (omitting one) to encrypt messages in pairs. Break the message into pairs (diagrams) and map them to the grid as opposite corners of a rectangle. Apply these rules:

1. If same letters (or single letter), add "X".
2. Same row, shift right.
3. Same column, shift down.
4. Different row/column, use the opposite corners of the rectangle on the same row.

**EXAMPLE:**

****

**ALGORITHM:**

STEP-1: Read the plain text from the user.

STEP-2: Read the keyword from the user.

STEP-3: Arrange the keyword without duplicates in a 5\*5 matrix in the row order and fill the remaining cells with missed out letters in alphabetical order. Note that ‘i’ and ‘j’ takes the same cell.

STEP-4: Group the plain text in pairs and match the corresponding corner letters by forming a rectangular grid.

STEP-5: Display the obtained cipher text.

**PROGRAM: (Playfair Cipher)**

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<ctype.h>

#define MX 5

void playfair(char ch1,char ch2, char key[MX][MX])

{

int i,j,w,x,y,z;

FILE \*out;

if((out=fopen("cipher.txt","a+"))==NULL)

{

printf("File Currupted.");

}

for(i=0;i<MX;i++)

{

for(j=0;j<MX;j++)

{

if(ch1==key[i][j])

{

w=i;

x=j;

}

else if(ch2==key[i][j])

{

y=i;

z=j;

}}}

//printf("%d%d %d%d",w,x,y,z);

if(w==y)

{

x=(x+1)%5;z=(z+1)%5;

printf("%c%c",key[w][x],key[y][z]);

fprintf(out, "%c%c",key[w][x],key[y][z]);

}

else if(x==z)

{

w=(w+1)%5;y=(y+1)%5;

printf("%c%c",key[w][x],key[y][z]);

fprintf(out, "%c%c",key[w][x],key[y][z]);

}

else

{

printf("%c%c",key[w][z],key[y][x]);

fprintf(out, "%c%c",key[w][z],key[y][x]);

}

fclose(out);

}

void main()

{

int i,j,k=0,l,m=0,n;

char key[MX][MX],keyminus[25],keystr[10],str[25]={0};

char

alpa[26]={'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'}

;

clrscr();

printf("\nEnter key:");

gets(keystr);

printf("\nEnter the plain text:");

gets(str);

n=strlen(keystr);

//convert the characters to uppertext

for (i=0; i<n; i++)

{

if(keystr[i]=='j')keystr[i]='i';

else if(keystr[i]=='J')keystr[i]='I';

keystr[i] = toupper(keystr[i]);

}

//convert all the characters of plaintext to uppertext

for (i=0; i<strlen(str); i++)

{

if(str[i]=='j')str[i]='i';

else if(str[i]=='J')str[i]='I';

str[i] = toupper(str[i]);

}

j=0;

for(i=0;i<26;i++)

{

for(k=0;k<n;k++)

{

if(keystr[k]==alpa[i])

break;

else if(alpa[i]=='J')

break;

}

if(k==n)

{

keyminus[j]=alpa[i];j++;

}

}

//construct key keymatrix

k=0;

for(i=0;i<MX;i++)

{

for(j=0;j<MX;j++)

{

if(k<n)

{

key[i][j]=keystr[k];

k++;}

else

{

key[i][j]=keyminus[m];m++;

}

printf("%c ",key[i][j]);

}

printf("\n");

}

printf("\n\nEntered text :%s\nCipher Text :",str);

for(i=0;i<strlen(str);i++)

{

if(str[i]=='J')str[i]='I';

if(str[i+1]=='\0')

playfair(str[i],'X',key);

else

{

if(str[i+1]=='J')str[i+1]='I';

if(str[i]==str[i+1])

playfair(str[i],'X',key);

else

{

playfair(str[i],str[i+1],key);i++;

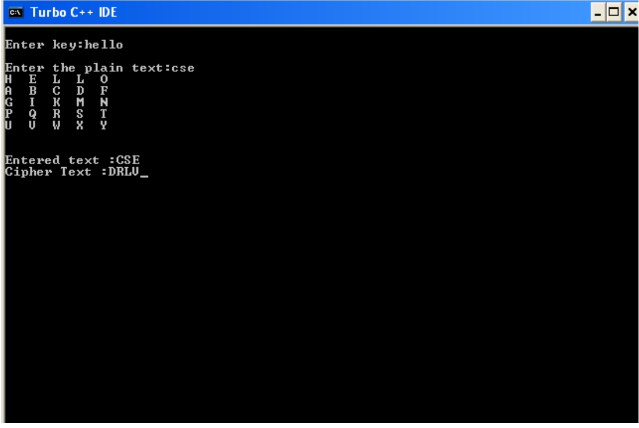
}}

}

getch();

}

**OUTPUT:**

****