PRACTICAL NO. 1

AIM: Implementation of Classical encryption techniques (Substitute and Transposition Cipher).

CAESAR CIPHER:

Code:

#include <stdio.h>

int main()

{

char msg[100],ch;

int i,key;

printf("Enter a message to encrypt: ");

gets(msg);

printf("Enter key: ");

scanf("%d",&key);

for(i=0;msg[i]!='\0';i++){

ch=msg[i];

if(ch>='a' && ch<='z'){

ch=ch+key;

if(ch>'z'){

ch=ch-'z'+'a'-1;

}

msg[i]=ch;

}

else if(ch>='A' && ch<='Z'){

ch=ch+key;

if(ch>'Z'){

ch=ch-'Z'+'A'-1;

}

msg[i]=ch;

}

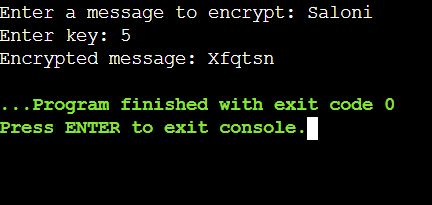
}

printf("Encrypted message: %s",msg);

return 0;

}

Output:



Polyalphabetic cipher technique:

Code:

#include<stdio.h>

#include<string.h>

void main()

{

char pt[20]={'\0'},ct[20]={'\0'},key[20]={'\0'},dt[20]={'\0'};

int i,j;

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

scanf("%s",pt);

printf("\nEnter the key: ");

scanf("%s",key);

//length of plaintext equal to length of the key

j=0;

for(i=strlen(key);i<strlen(pt);i++)

{

if(j==strlen(key))

{

j=0;

}

key[i]=key[j];

j++;

}

printf("\n New key is: %s",key);

//converting plain text to cipher text (encryption)

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

{

ct[i]=(((pt[i]-97)+(key[i]-97))%26)+97;

}

printf("\n\nCipher text is: %s",ct);

//converting cipher text to plain text (decryption)

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

{

if(ct[i]<key[i])

{

dt[i]=26+((ct[i]-97)-(key[i]-97))+97;

}

else

dt[i]=(((ct[i]-97)-(key[i]-97))%26)+97;

}

printf("\n\nPlain text is:%s",dt);

}

Output:

